

Aalto-yliopisto
Perustieteiden korkeakoulu
Informaatioverkostojen koulutusohjelma

Antti Vuorela

Mobiilin uutispalvelun teknologiastrategia:

Web-, natiivi- ja hybriditeknologioiden vertailu mobiilin uutispalvelun vaatimusten näkökulmasta

Diplomityö
Espoo, 26. huhtikuuta 2016

Valvoja: Professori Heikki Saikkonen
Ohjaaja: Sami Vihavainen, TkT

Aalto-yliopisto
 Perustieteiden korkeakoulu
 Informaatioverkostojen koulutusohjelma

DIPLOMITYÖN
 TIIVISTELMÄ

Tekijä:	Antti Vuorela		
Työn nimi:	Mobiilin uutispalvelun teknologiastrategia: Web-, natiivi- ja hybriditeknologioiden vertailu mobiilin uutispalvelun vaatimusten näkökulmasta		
Päiväys:	26. huhtikuuta 2016	Sivumäärä:	viii + 80
Pääaine:	Media	Koodi:	IL3009
Valvoja:	Professori Heikki Saikkonen		
Ohjaaja:	Sami Vihavainen, TkT		
<p>Älykkäät mobiililaitteet ovat viime vuosina mullistaneet ihmisten tavan käyttää digitaalisia palveluita. Myös uutispalveluiden käyttö mobiililaitteissa on suosittua ja media-alan toimijat panostavat mobiilipalveluiden kehittämiseen. Uutispalvelut haluavat palvella mahdollisimman laajaa käyttäjäjoukkoa, mutta erilaisten mobiililaitteiden ja -alustojen suuri määrä aiheuttaa haasteita kehitystyölle.</p> <p>Tämän tutkimuksen tavoitteena on selvittää minkälaisilla mobiiliteknologioilla pystytään parhaiten vastaamaan uutispalveluiden erityisvaatimuksiin. Ensinnäkin tunnistetaan uutispalveluiden vaatimukset käytettävillä mobiiliteknologioille. Seuraavaksi tutkitaan kuinka kolme toteutustapaa: web-sivustot, natiivisovellukset ja näiden yhdistelmä, hybridi-sovellukset, vastaavat löydettyihin vaatimuksiin.</p> <p>Tutkimus on tapaustutkimus Ilta-Sanomien mobiileista uutispalveluista. Tutkimuksen päämateriaalina käytetään kymmenen Ilta-Sanomien mobiilipalveluiden kehityksen parissa työskennelleen asiantuntijan haastatteluita. Materiaali analysoidaan käyttäen grounded theory -metodia.</p> <p>Tutkimuksen tuloksena tunnistetaan viisi päävaatimusta mobiilille uutispalvelulle ja sen toteutusteknologioille: <i>yhteneväisyys, muokattavuus, käyttökokemus ja ulkoasu, kaupalliset tavoitteet</i> sekä <i>nopeus ja luotettavuus</i>. Natiivisovellusten ja web-sivuston todetaan olevan uutispalvelun jakelukanavina erilaisia, mutta toisiinsa täydentäviä ja tukevia. Hybridi-sovelluksien avulla todetaan olevan osittain mahdollista tuoda web-teknologioiden etuja natiivisovelluksiin ja siten vastata paremmin joihinkin uutispalvelun vaatimuksiin.</p> <p>Johtopäätöksenä esitetään <i>uutispalvelun mobiiliteknologiastrategia</i>, joka muodostuu yleisistä periaatteista sekä natiivisovellusten toteutusvaihtoehdoista. Tärkein yleinen periaate on, että mahdollisimman suuri osa sisällön rakenteesta ja käyttöliittymälogiikasta kannattaa määritellä palvelun taustajärjestelmissä eikä eri jakelukanavissa.</p>			
Asiasanat:	uutispalvelu, mobiili, web, natiivi, hybridi, mobiilipalvelu, mobiilisovellus, mobiililaitte, ohjelmistokehitys		
Kieli:	Suomi		

Aalto University
 School of Science
 Information Networks Degree Program

ABSTRACT OF
 MASTER'S THESIS

Author:	Antti Vuorela		
Title:	Mobile technology strategy for a news service: Comparing web, native and hybrid technologies based on the requirements of a mobile news service		
Date:	April 26, 2016	Pages:	viii + 80
Major:	Media	Code:	IL3009
Supervisor:	Professor Heikki Saikkonen		
Advisor:	Sami Vihavainen, D.Sc. (Tech.)		
<p>Smartphones and other mobile devices have revolutionized the way people use digital services. News services are popular on mobile devices and media companies are focusing more and more on developing mobile services. News services aim to serve a wide audience, but the large variety of different mobile devices and platforms presents a challenge for the development of mobile services.</p> <p>The goal of this thesis is to evaluate, what type of mobile technologies best fulfill the specific requirements of news services. First, news services' requirements for mobile technologies are identified. Then three implementation methods – websites, native applications and the combination of these, hybrid applications – are evaluated against the identified requirements.</p> <p>The research is based on a case study of Ilta-Sanomat mobile news service. The main material of the study consists of interviews of ten experts who have worked on developing Ilta-Sanomat mobile services. Material is analyzed using Grounded Theory -method.</p> <p>As the result of the study, five main requirements for a mobile news service and implementation technologies are identified: <i>consistency</i>, <i>editability</i>, <i>user experience and design</i>, <i>business targets</i> and finally <i>speed and reliability</i>. Native applications and websites are found to be totally separate distribution channels for a news service, but complementing and supporting each other. Hybrid applications are found to be partly able to bring the benefits of web-technologies to native applications and thus better fulfill some of the news service requirements.</p> <p>As a conclusion, <i>mobile technology strategy for a news service</i> is presented. The strategy consists of general principles, as well as implementation options for native applications. The most important general principle is that as much of the content structure and user interface logic as possible should be defined in the backend systems, not in the different distribution channels.</p>			
Keywords:	news service, mobile, web, native, hybrid, mobile service, mobile application, mobile device, software development		
Language:	Finnish		

Alkusanat

Tämä työ on tehty yhteistyössä työnantajani Futuricen sekä asiakkaamme Ilta-Sanomien kanssa. Suuret kiitokset työn mahdollistamisesta kuuluvat Ilta-Sanomien edustajille Timo Rinteelle, Nicklas Koskelle ja Teemu Hauhialle. Työskentely Suomen suurimman uutispalvelun parissa on tarjonnut minulle paljon mielenkiintoisia haasteita ja uusien asioiden oppimista.

Työnantajani Futuricea haluan kiittää mahdollisuudesta osallistua diplomityöleirille. Sami Vihavaista haluan kiittää erittäin asiantuntevasta ohjauksesta diplomityöleirillä. Iso kiitos kuuluu myös kollegoilleni, erityisesti kaikille Ilta-Sanomien tiimissä työskennelleille ja muille työni tekemistä tukeneille. Olen etuoikeutettu saadessani työskennellä Euroopan parhaassa työpaikassa.

Työn valvojaa professori Heikki Saikkosta haluan kiittää pitkäjänteisestä yhteistyöstä ja lukuisista hyvistä kysymyksistä diplomityöprosessin varrella.

Vanhempiani haluan kiittää kaikesta tuesta mitä olen lapsuudesta alkaen saanut koulunkäyntiini ja opiskeluihini.

Eniten haluan kuitenkin kiittää vaimoani Hannelea valtavasta tuesta, rakkaudesta ja kärsivällisyydestä. Ilman Hannelen jatkuvaa kannustusta tämä työ ei todennäköisesti olisi koskaan valmistunut. Nyt voimme jatkaa eteenpäin kohti uusia seikkailuja.

Espoossa, 26.4.2016

Antti Vuorela

Sisältö

1	Johdanto	1
1.1	Motivaatio	1
1.2	Työn tavoitteet ja rajausta	1
1.3	Tutkimuskysymykset	2
1.4	Työn rakenne	3
2	Aikaisempi tutkimus	4
2.1	Mobiililaitteet ja -palvelut	4
2.1.1	Mobiilialustat ja ekosysteemit	4
2.2	Mobiilipalveluiden kehittäminen	6
2.2.1	Natiiviteknologiat	6
2.2.2	Web-teknologiat	7
2.2.3	Hybridi-teknologiat	8
2.3	Mobiilipalvelun teknologiastrategia	9
2.3.1	Teknologiavaihtoehtojen arviointimallit	10
2.3.2	Teknologiavaihtoehtojen vertailu ja suositukset	11
2.4	Käyttäjätilastot	12
2.4.1	Median käytön muutokset	13
2.4.2	Mobiilipalveluiden käyttö	13
2.5	Yhteenvedo aikaisemmista tutkimuksista	14
3	Case: Ilta-Sanomien mobiilit uutispalvelut	15
3.1	Ilta-Sanomat	15

3.2	Ilta-Sanomien mobiilipalvelut	16
3.3	Ilta-Sanomien uutispalvelu	19
3.3.1	Natiivit uutissovellukset	20
3.3.2	Mobiilisivusto	20
3.3.3	Responsiivinen web-sivusto	20
4	Materiaalit ja metodit	21
4.1	Tutkimusmenetelmät	21
4.2	Haastattelut	22
4.2.1	Analyysimenetelmä: Grounded theory	23
4.2.2	Haastattelujen analyysiprosessi	23
4.3	Hybridi-prototyypit	26
5	Tulokset	28
5.1	Uutispalvelun vaatimukset mobiiliteknologioille	28
5.1.1	Sisällön yhteneväisyys, toimivuus ja hallittavuus	29
5.1.2	Muokattavuus: nopeat ja hallitut muutokset	31
5.1.3	Käyttökokemus ja ulkoasu: tuntuu brändin mukaiselta ja noudattaa alustojen konventioita	32
5.1.4	Kaupalliset tavoitteet: mainosinventaarin kasvattaminen laajalla käyttäjäpohjalla, säännöllisellä käytöllä ja uusilla mainostyypeillä	32
5.1.5	Nopeus ja luotettavuus: sisällön julkaiseminen nopeaa, palvelu toimii nopeasti ja luotettavasti kaikissa tilanteissa	34
5.1.6	Personointi: tulevaisuuden vaatimus	35
5.1.7	Yhteenveto	36
5.2	Miten web- ja natiiviteknologiat vastaavat uutispalvelun vaatimuksiin?	38
5.2.1	Natiivisovellusten ja web-sivuston erot jakelukanavina ja käyttötavoissa	38
5.2.2	Sisällön yhteneväisyyden ja toimivuuden ylläpito on ongelma natiivisovelluksissa	41

5.2.3	Web-sivuston muutosten tekeminen on nopeampaa ja päivitykset hallittuja	44
5.2.4	Käyttökokemuksessa natiivisovellukset ovat alustan mukaisia	47
5.2.5	Kaupalliset tavoitteet: natiivisovellukset tuottoisampia, uusissa mainostyypeissä ongelmia	48
5.2.6	Natiivisovellukset ovat nopeampia ja luotettavampia . .	50
5.2.7	Yhteenveto	53
5.3	Miten hybridi-sovellukset vastaavat uutispalvelun vaatimuksiin?	55
5.3.1	Sisällön yhteneväisyys ja hallittavuus merkittävin parannus natiivisovelluksiin	55
5.3.2	Nopeat muutokset mahdollisia hybridinäkymiin, mutta kokonaisuuden hallintaan on kiinnitettävä huomiota	56
5.3.3	Käyttökokemuksen takaaminen ja alustakohtaisen ulkoasun toteuttaminen haasteellista	56
5.3.4	Vaikutukset kaupallisiin tavoitteisiin epäselviä	57
5.3.5	Nopeus ja luotettavuus	57
5.3.6	Yhteenveto	59
6	Mobiiliteknologiastrategia	61
6.1	Strategian yleiset periaatteet	61
6.1.1	Sisällön rakenteen ja käyttöliittymälogiikan määrittely taustajärjestelmissä	61
6.1.2	Sekä Web-sivusto, että alustojen natiivisovellukset tarvitaan	62
6.1.3	Jaettu ohjelmalogiikka palveluiden ja alustojen välillä .	62
6.1.4	Elinkaariajattelu: natiivisovellusten sulkeminen, päivitysilmoitukset ja rajapintojen versiointi	63
6.1.5	Nopeuden ja laadun korostaminen	64
6.2	Natiivisovellusten toteutusvaihtoehdot	65
6.2.1	Hybridi mahdollistaa yhteneväisen sisällön ja ulkoasun, helpon hallittavuuden sekä tehokkaan kehittämisen usealle alustalle	65

6.2.2	Natiivi mahdollistaa alustakohtaisen mukauttamisen ja käyttökokemuksen	65
6.3	Yhteenveto	66
7	Keskustelu ja arviointi	67
7.1	Työn tulosten arviointi	67
7.2	Mobiiliteknologioiden ja markkinatilanteen kehittyminen . . .	68
7.3	Tulevat tutkimukset	69
8	Yhteenveto	70
	Lähteet	74
A	Haastattelukysymykset	78
A.1	Kysymykset kaikille	78
A.2	Lisäkysymykset Sanoman edustajille	79
A.3	Lisäkysymykset kehittäjille	80

Luku 1

Johdanto

1.1 Motivaatio

Älykkäät mobiililaitteet ovat viime vuosina mullistaneet ihmisten tavan käyttää digitaalisia palveluita. Myös uutispalveluiden käyttö mobiililaitteissa on erittäin suosittua ja media-alan toimijat panostavatkin vauhdilla mobiilipalveluiden kehittämiseen.

Uutispalvelut haluavat palvella mahdollisimman laajaa käyttäjäjoukkoa, mutta erilaisten mobiililaitteiden ja -alustojen suuri määrä aiheuttaa haasteita nopealle kehitystyölle. Lisäksi uutispalveluilla vaikuttaa olevan monia erityisiä vaatimuksia, kuten nopeasti muuttuva sisältö ja sen pitäminen yhteneväisenä kaikissa eri jakelukanavissa.

Tulevaisuudessa mobiilipalveluiden rooli uutisten jakelukanavina ja kehitystyön tahti tulee entisestään kasvamaan. Mobiilipalveluiden kehittämisen pitäisi olla entistä ketterämpää ja isojaakin muutoksia olisi hyvä pystyä tekemään nopeasti siten, että kaikkia käyttäjiä palvellaan mahdollisimman hyvin. Nämä tavoitteet asettavat isoja haasteita mobiilipalveluiden kehitystyölle ja käytettäville teknologioille.

1.2 Työn tavoitteet ja rajaus

Tämän tutkimuksen tavoitteena on selvittää minkälaisilla mobiiliteknologioilla pystytään parhaiten vastaamaan uutispalvelun erityisvaatimuksiin. Tutkimuksen tarkastelu on rajattu toimituksellisen, laajalle lukijapohjalle suunnatun ja jatkuvasti päivittyvän uutispalvelun kontekstiin. Tarkoitus ei

ole löytää geneerisesti kaikkia mobiilipalveluita koskevia tuloksia vaan mahdollisimman konkreettisia vastauksia uutispalveluiden tapaukseen.

Aikaisemmissa aihepiirin tutkimuksissa on tutkittu ja vertailtu erilaisia tekniikoita mobiilipalveluiden toteutukseen, mutta erityisesti uutispalvelun kontekstissa tehtyjä tutkimuksia ei löytynyt. Aikaisempien tutkimusten lopputulos on yleisesti ollut, että teknologiavalinta riippuu kehitettävän palvelun erityispiirteistä ja vaatimuksista. Aikaisempaa tutkimusta käsitellään laajemmin luvussa 2.

Tämän työn tavoitteena onkin tunnistaa mobiilin uutispalvelun erityispiirteet sekä vaatimukset käytettäville mobiiliteknologioille ja vertailla eri toteutusvaihtoehtoja niiden pohjalta. Työssä vertaillaan kolmea tapaa toteuttaa uutispalvelu mobiililaitteisiin: mobiililaitteille optimoitu *web-sivusto*, alustakohtaiset *natiivisovellukset* ja kahta edellistä tekniikkaa yhdistelevät *hybridisovellukset*.

Työn johtopäätöksenä muodostetaan mobiilin uutispalvelun teknologiastrategia, jossa annetaan suosituksia mobiilipalveluiden toteuttamiseen uutispalveluille.

Työ on tapaustutkimus, jonka empiirisenä tapauksena tarkastellaan Ilta-Sanomien uutispalvelua. Tutkimuksen materiaalina on käytetty Ilta-Sanomien uutispalveluiden kehittämisessä kertynyttä tietotaitoa, joka on kerätty haastattelemalla kymmentä palveluiden kehityksen parissa työskennellyttä asiantuntijaa. Lisäksi materiaalina on käytetty Ilta-Sanomien uutissovelluksista tehtyjä hybridi-prototyyppisiä ja niistä saatuja kokemuksia. Työn materiaalit ja metodit on esitelty tarkemmin kappaleessa 4.

1.3 Tutkimuskysymykset

Edellä kuvattujen tavoitteiden täyttämiseksi työssä pyritään vastaamaan kolmeen tutkimuskysymykseen:

1. Minkälaisia vaatimuksia uutispalvelulla on käytettäville mobiiliteknologioille?
2. Miten web- ja natiiviteknologiat vastaavat uutispalvelun vaatimuksiin ja minkälaisia ongelmia niiden kanssa nykyisellään on?
3. Voiko hybridi-sovelluksilla yhdistää web-sivuston ja natiivisovellusten hyvät puolet ja vastata entistä paremmin mobiilin uutispalvelun vaatimuksiin?

1.4 Työn rakenne

Työ alkaa luvussa 2 työn aihepiiristä tehdyn aikaisemman tutkimuksen käsittelyllä. Luvussa 3 esitellään työn tapaustutkimus, Ilta-Sanomien mobiilit uutispalvelut. Luvussa 4 puolestaan käydään läpi työssä käytetyt materiaalit ja niiden analysoimiseen käytetyt metodit.

Luvussa 5 esitellään kattavasti työn tulokset tutkimuskysymyksittäin ja se muodostaakin työn isoimman ja merkittävimmän osan. Ensin käydään läpi minkälaisia vaatimuksia uutispalveluilla on käytettäville mobiiliteknologioille. Seuraavaksi esitellään tulokset web-sivustoista ja natiivisovelluksista uutispalveluiden jakelukanavina. Lopuksi käydään läpi voisiko hybridi-sovelluksella web- ja natiivitekniikoita yhdistelemällä vastata entistä paremmin uutispalvelun vaatimuksiin.

Luvussa 6 muodostetaan tulosten pohjalta johtopäätökset ja esitys mobiilin uutispalvelun teknologiastrategiaksi. Luvussa 7 puolestaan arvioidaan työn tuloksia ja keskustellaan mobiiliteknologioiden kehittymisestä. Lopuksi luvussa 8 tehdään yhteenveto työn tuloksista ja johtopäätöksistä.

Luku 2

Aikaisempi tutkimus

Tässä luvussa käydään läpi työn aihepiiristä tehtyä aikaisempaa tutkimusta. Ensimmäiseksi käsitellään mobiililaitteisiin ja -palveluihin liittyviä yleisiä konsepteja. Toiseksi käydään läpi eri teknologioita joita mobiilipalveluiden kehityksessä voidaan käyttää. Kolmanneksi tutustutaan teknologiavaihtoehtojen vertailuihin ja suosituksiin, joita aikaisemmissa tutkimuksissa on annettu. Lopuksi käydään läpi tämän työn aihepiirin kannalta olennaisia käyttäjätilastoja.

2.1 Mobiililaitteet ja -palvelut

Tässä tutkimuksessa *mobiililaitteilla* viitataan moderneihin älypuhelimiin ja tablet-tietokoneisiin. *Mobiilipalveluilla* puolestaan tarkoitetaan näille laitteille suunnattuja digitaalisia palveluita.

2.1.1 Mobiilialustat ja ekosysteemit

Mobiililaitteiden käyttöjärjestelmien muodostamia laiteperheitä kutsutaan *mobiilialustoiksi* (engl. *platform*). Tietyn mobiilialustan laitteet käyttävät samaa käyttöjärjestelmää ja niillä toimii samat ohjelmistot. Suosituimpia mobiilialustoja ovat työn kirjoittamishetkellä Googlen *Android*, Applen *iOS* sekä Microsoftin *Windows Phone*. (Winter, 2014)

Mobiilialustat puolestaan muodostavat *ekosysteemejä*. Ekosysteemit koostuvat alustalle palveluja tuottavista tahoista ja alustan ympärillä toimivasta liiketoiminnasta. Ekosysteemille ei ole yhtä yleisesti todettua määritelmää.

Campbell ja Ahmed (2011) ovat vertailleet eri määritelmiä ja ovat niiden perusteella määritelleet ekosysteemin seuraavasti:

”[Software ecosystem is] a cluster of actors (individuals or organizations) employed in the development of services or software for a common market and sharing a common technological framework”

Basole ja Karla (2012) ovat tutkineet mobiilipalveluiden ekosysteemejä ja niissä tapahtuvia arvoketjuja. He esittävät, että mobiilipalveluiden ekosysteemi muodostuu neljästä päätoimijasta: *alustatoimittajat*, *laitevalmistajat*, *verkko-operaattorit* ja *sovelluskehittäjät*. Yhdessä nämä neljä toimijaa muodostavat arvoverkoston, joka tuottaa lisäarvoa toisilleen.

Keskeinen osa mobiilialustojen ekosysteemejä ovat nykyään *ohjelmistokaupat*, joiden kautta sovelluskehittäjät voivat tarjota ja myydä natiivisovelluksia käyttäjien mobiililaitteisiin. Ohjelmistokauppoja käsitellään tarkemmin kappaleessa 2.2.1. Basole ja Karla (2012) esittävät, että ohjelmistokaupat ovat perusteellisesti muuttaneet mobiiliekosysteemien toimijoiden voima-suhteita. Ennen ohjelmistokauppoja verkko-operaattoreilla oli huomattavasti merkittävämpi rooli ekosysteemien toiminnassa ja ne hallitsivat sisällön jakelua käyttäjille.

Winter (2014) on lisensointityössään tutkinut kolmea suurinta mobiilialan ekosysteemiä: Applen iOS, Googlen Android ja Microsoftin Windows Phone. Winter on analysoinut ekosysteemejä ja tunnistanut kahdeksan yhteistä menestystekijää, jotka määrittävät ekosysteemien ja alustojen suosiota. Näihin menestystekijöihin kuuluu esimerkiksi se, että kaikkien kolmen ekosysteemin mobiilialustat tarjoavat riittävästi avoimia sovellusrajapintoja, jotta laitevalmistajat ja sovelluskehittäjät pystyvät toteuttamaan monipuolisesti erottuvia tuotteita ja sovelluksia.

Karhu et al. (2014) ovat myös tutkineet suurimpia mobiiliekosysteemejä ja pyrkinet muodostamaan mitattavan mallin ekosysteemien sisäisen kilpailun ja yhteistyön analysointiin. Mallin avulla he ovat arvioineet mobiiliekosysteemien strategioita. Heidän analyysinsä esittää, että eri ekosysteemeissä on valittu hyvin erilaiset strategiat ekosysteemien toimijoiden väliseen yhteistyöhön. Applen iOS-ekosysteemi on suljettu, kun taas Googlen Android-ekosysteemi painottaa eniten yhteistyötä toimijoiden välillä.

2.2 Mobiilipalveluiden kehittäminen

Mobiilipalveluiden kehittäminen tarkoittaa mobiilikäyttöön tarkoitetun palvelun suunnittelua ja teknistä toteutusta. *Jakelukanava* puolestaan on tapa, jolla palvelu välitetään loppukäyttäjien mobiililaitteisiin. Tämän tutkimuksen puitteissa käsitellään kahta mobiilipalvelun jakelukanavaa: alustan sovel-luskaupasta asennettavia natiivisovelluksia sekä web-selaimella käytettäviä mobiililaitteille optimoituja web-palveluita.

Tutkimuksen tarkastelusta on jätetty pois muut mahdolliset mobiilipalveluiden jakelukanavat, esimerkiksi SMS-viesteillä käytettävät palvelut, jotka eivät modernien mobiililaitteiden tapauksessa ole enää relevantteja.

Natiivisovelluksia ja web-sivustoja jakelukanavina ja teknisinä toteutus-malleina tutkitaan tarkemmin kappaleissa 2.2.1 ja 2.2.2. Näiden mallien väliin asettuu kuitenkin vielä vaihtoehto, jossa web- ja natiivitekniikoita käytetään yhdessä. Tätä lähestymistapaa kutsutaan *hybridi-teknologiaksi* ja sitä käsitellään tarkemmin kappaleessa 2.2.3.

2.2.1 Natiivitekniologiat

Natiivitekniologioilla tarkoitetaan mobiililaitteeseen asennettavan *mobiili-sovelluksen* toteuttamista mobiilialustan tukemilla ohjelmointikielillä ja työkaluilla. Natiivisovellukset kehitetään käyttäen alustatoimittajan tarjoamia alustakohtaisia rajanpintoja sekä komponentteja.

Eri alustoilla on käytössä hyvin erilaisia ohjelmointikieliä ja -paradigmoja natiivisovellusten kehittämiseen. Sovellusten kehittäminen eri alustoille vaatii näin ollen hyvin erilaisia ohjelmistokehitystaitoja. Kuvassa 2.1 on esitetty eri alustoilla vaadittavia teknologiataitoja. (Charland and Leroux, 2011)

Alustojen ohjelmistokaupat natiivisovellusten jakelukanavana

Natiivisovellukset jaetaan loppukäyttäjille useimmiten alustojen ohjelmis-tokauppojen kautta. Useilla alustoilla, kuten Applen iOS:lla ja Microsoftin Windows Phonella, alustavalmistajan viralliset ohjelmistokaupat ovat ainoa sallittu tapa toimittaa natiivisovellus loppukäyttäjälle. Googlen Androidissa sen sijaan sovellusten asentaminen suoraan kehittäjältä tai muista kilpaile-vista ohjelmistokaupoista on myös mahdollista.

Juntunen et al. (2013) esittää, että yksi natiivisovelluskauppojen suurimmis-ta hyödyistä on mahdollisuus sovellusten helppoon kaupallistamiseen; joko

Required skill sets for nine mobile OSs.	
Mobile OS Type	Skill Set Required
Apple iOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Window 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Kuva 2.1: Natiivisovellusten kehittämisessä vaadittavia teknologioita (Charland and Leroux, 2011)

laskuttamalla sovellusten lataamisesta tai myymällä sovelluksen sisäisiä osia (engl. *in-app-purchases*). Kääntöpuolena sovelluskaupan omistaja ottaa provision saadusta tuotosta - yleisesti 30 prosenttia hinnasta. Lisäksi sovelluskaupoissa on säännöt, jotka sovellusten tulee täyttää päästäkseen jakeluun kaupassa. Winter (2014) mukaan erityisesti Applen ekosysteemissä sovellusten kontrolli ja säännöt¹ ovat hyvin tiukkoja.

2.2.2 Web-teknologiat

Web-teknologioilla tarkoitetaan palvelun kehittämistä käyttäen HTML-kuvauskieltä ja siihen liittyviä tekniikoita, kuten JavaScript-ohjelmointikieltä ja CSS-tyylitiedostoja. Näillä teknologioilla tehtyjä palveluita käytetään *web-selaimessa*, joka muuttaa HTML-kuvauksen ja siihen liittyvät toiminnallisuudet sekä tyylit käyttäjälle näkyväksi käyttöliittymäksi.

Modernien HTML, JavaScript ja CSS -tekniikoiden hyödyntämiseen viitataan termillä *HTML5-teknologiat*. HTML5 on HTML-kuvauskielen uusin standardi, mutta termiä käytetään yleisesti myös yläkäsitteenä HTML-kieleen liittyvistä teknologioista. Termillä ei siis yleensä tarkoiteta mitään yksittäistä teknologiaa, vaan kokoelmaa erilaisia jatkuvasti kehittyviä tek-

¹<https://developer.apple.com/app-store/review/guidelines/>

niikoita. Karkeasti jaoteltuna HTML-kieltä käytetään sisällön kuvaamiseen, CSS-tyylitiedostoja ulkoasun määrittelyyn ja JavaScript-kieltä selaimessa tapahtuviin toiminnallisuuksiin. (Juntunen et al., 2013)

Web-tekniologiat mobiilipalveluiden kehittämisessä

Puputti (2012) on tutkinut diplomityössään moderneihin web-tekniikoihin pohjautuvan mobiilipalvelun toteuttamista. Puputti esittää, että suurimmat haasteet web-tekniikoiden käytössä mobiililaitteissa liittyvät eri näyttökokojen ja resoluutioiden tukemiseen, epäluotettavien verkkoyhteyksien ja offline-tilan hallintaan sekä käyttöliittymän suorituskykyyn erityisesti animaatioiden kanssa. Hän kuitenkin toteaa, että moderneja HTML5-tekniikoita hyödyntämällä ja suunnittelemalla sovelluksen arkkitehtuuri huolellisesti nämä ongelmat voidaan pääosin välttää.

Nieminen (2015) on diplomityössään tutkinut tarkemmin epäluotettavien verkkoyhteyksien ja offline-tilan toteuttamista mobiilissa web-sovelluksessa. Nieminen esittää, että yhteysongelmiin ja offline-tilan toteuttamiseen tulee varautua jo sovelluksen arkkitehtuuria suunniteltaessa. Vaikka offline-tukea ei toteutettaisikaan, varautumalla epäluotettaviin verkkoyhteyksiin voidaan parantaa web-sovelluksen käyttökokemusta huomattavasti.

2.2.3 Hybridi-tekniologiat

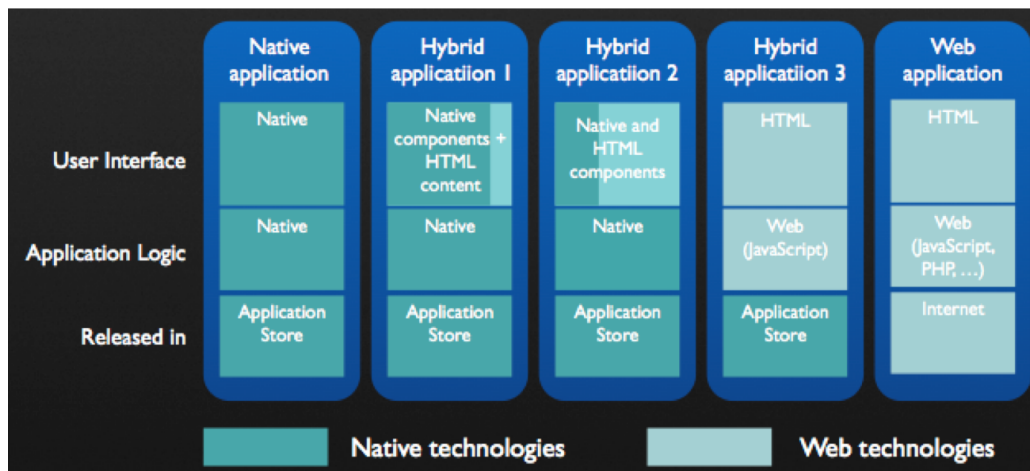
Hybridi-tekniologioissa yhdistetään natiivi- ja web-tekniologioita. Natiivisovelluksen sisällä osa toiminallisuuksista toteutetaan hyödyntäen web-tekniologioita. Käytännössä natiivisovellukseen upotetaan *web-näkymä*, jossa voidaan esittää web-tekniologioilla tehtyä sisältöä.

Mobiilialustan sekä käyttäjän näkökulmasta hybridi-sovellus vastaa konseptina natiivisovellusta, ainoastaan toteutustekniologia on eri. Hybridi-sovelluksen jakelukanava on vastaava kuin natiivisovelluksissa, eli se tulee asentaa alustan ohjelmistokaupasta.

Hybridi-lähestymistavalla tehdyissä sovelluksissa on huomattavasti eroja sen mukaan kuinka suuri osa osa sovelluksesta on toteutettu web-tekniikoita hyödyntäen. Tämä tulee huomioida vertailtaessa hybridi-tekniologioilla tehtyjä sovelluksia. Kokemukset hybridi-sovelluksista eroavat paljon sen mukaan, mihin web-tekniikoita on natiivisovelluksessa käytetty.

Zvonkov (2013) jaottelee hybridi-sovellukset kolmeen tasoon sen mukaan kuinka paljon niissä hyödynnetään web-tekniikoita. Zvonkovin jaottelu mo-

biilipalveluiden jakelukanavista on esitetty kuvassa 2.2.



Kuva 2.2: Zvonkovin (2013) malli mobiilipalvelun teknologiavaihtoehdoista

Zvonkovin mallissa ensimmäisen tason hybridisovelluksissa web-tekniikoita on hyödynnetään vain sisällön esittämiseen ja tyyliin. Toisen tason hybridisovelluksissa web-tekniikoita käytetään myös käyttöliittymäkomponenttien toteuttamiseen, sovelluslogiikan ollessa edelleen natiivia. Kolmannen tason hybridisovelluksissa koko käyttöliittymä ja sovelluslogiikka on toteutettu web-tekniikoilla ja natiivisovellus on vain kevyt kehys web-sovelluksen ympärillä.

Savuoja (2015) on diplomityössään kartoittanut asioita, joita tulee ottaa huomioon kehitettäessä hybridisovelluksia mobiililaitteille. Savuoja esittää, että hybridisovelluksen sisälle paketoitavaa web-sovellusta kehitettäessä on huomioitava mobiilialustojen erityispiirteet ja rajoitteet. Käyttöliittymä tulee suunnitella siten, että se on luonteva jokaisella julkaisualustalla. Savuoja toteaa, että sama web-sovellus voidaan tehokkaasti jaella loppukäyttäjille sekä hybridisovelluksena että tavallisena web-sivustona. Näiden välillä on mahdollista uudelleenkäyttää lähes kaikkia sovelluksen osia.

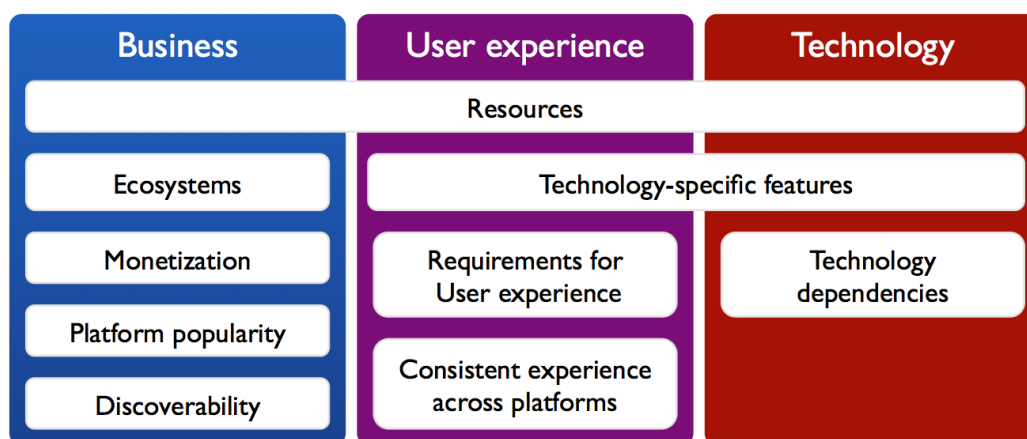
2.3 Mobiilipalvelun teknologiastrategia

Mobiilipalvelun teknologiastrategialla tarkoitetaan tämän tutkimuksen kontekstissa mobiilipalvelun jakelukanavien ja toteutusteknologioiden valitsemista. Ensimmäisessä alakappaleissa käydään läpi minkälaisia teknologia-vaihtoehtojen arviointimalleja aikaisemmissa tutkimuksissa on käytetty. Sen

jälkeen esitellään aikaisemmissa töissä tehtyjä vertailuja eri toteutustavoista. Lopuksi käydään läpi minkälaisia suosituksia teknologiavalintaan on aikaisemmin annettu.

2.3.1 Teknologiavaihtoehtojen arviointimallit

Zvonkov (2013) muodostaa diplomityössään mallin teknologiavalintaan vaikuttavista elementeistä. Zvonkovin mukaan menestyksellinen mobiilipalvelu muodostuu kolmesta komponentista: menestyksellisestä liiketoimintamallista, hyvästä käyttökokemuksesta ja korkealaatuisesta teknisestä toteutuksesta. Nämä kolme komponenttia muodostavat kategoriat, joihin teknologiavalintaan vaikuttavat elementit jakautuvat. Kuvassa 2.3 on esitetty tarkemmin Zvonkovin jaottelu.



Kuva 2.3: Zvonkovin (2013) malli teknologiavalintaan vaikuttavista elementeistä

Ottka (2015) puolestaan on tutkinut natiivi-, web- ja hybridisovellusten valintatilannetta tietyssä käyttötapauksessa: teollisuuskäyttöön tarkoitetuissa monialustaisissa sovelluksissa. Hän on muodostanut arviointimallin toteutusteknologian valintatilanteeseen. Ottkan malli muodostuu 12 osa-alueesta, jotka jakautuvat kolmeen näkökulmaan: Loppukäyttäjän näkökulman osa-alueet ovat *käyttöliittymän yhteneväisyys kohdealustan kanssa*, *interaktiot ja responsiivisuus*, *tuetut alustat* ja *energiankäyttö*. Kehittäjän näkökulman osa-alueet ovat *jaettu koodi*, *pääsy alustan ominaisuuksiin*, *kykyjen saatavuus* ja *suunnittelun helppous*. Liiketoimintanäkökulman osa-alueet puolestaan ovat *kustannukset*, *pitkän aikavälin soveltuvuus*, *julkaisu ja jakelu* sekä *turvallisuus*.

2.3.2 Teknologiavaihtoehtojen vertailu ja suositukset

Juntunen et al. (2013) ovat tutkineet HTML5-tekniikoiden käyttöä mobiililaitteissa ja vertailleet niitä natiivitekniikoihin. He esittävät, että natiivisovellusten kehityksessä ja jakelumallissa on ongelmia, joihin web-tekniikat osittain vastaavat. Mobiilialustat ja -laitteet ovat fragmentoituneet ja natiivisovellukset ovat sidottuja tiettyyn alustaan. Tämän johdosta isoa käyttäjäkuntaa natiivisovelluksilla tavoittelevan mobiilipalvelun täytyy tehdä sovellukset erikseen usealle eri mobiilialustalle, mikä lisää merkittävästi kehitystyön kustannuksia ja kestoja.

Juntunen et al. toteavat, että tärkeä etu web-tekniikoiden käyttämisestä on yhteensopivuus usean mobiilialustan kanssa. Tämä mahdollistaa mobiilipalveluiden helpomman kehittämisen usealle alustalle ja vähentää mobiilialustojen fragmentaation ongelmaa. Natiivisovellusten etuja puolestaan ovat mahdollisuus hyödyntää natiiveja rajapintoja sekä web-sovelluksia parempi käytettävyys ja suorituskyky. He esittävät, että web-sovellukset ovat kustannustehokas tapa tuottaa palvelu isolle käyttäjäjoukolle usealla alustalla.

Fling (2009) esittää, että natiivisovellus kannattaa tehdä jos sovelluksen tarvitsee hyödyntää alustan ominaisuuksia, joihin web-selaimella ei ole pääsyä. Muutoin hän suosittelee mobiiliin web-sovelluksen tekemistä. Flingin mukaan web-sovellukset ovat pitkällä aikavälillä ainoa järkevä alusta mobiilipalveluiden tarjoamiseen.

Webin keksijä Berners-Lee (2010) puhuu vahvasti avoimen webin ideologian ja periaatteiden puolesta. Hän moittii natiivisovellusten suosiota häiritseväksi, sillä se luo suljettuja maailmoja avoimen webin ulkopuolelle. Berners-Lee moittii sovelluksia linkattavuuden puutteesta ja sen sivuvaikutuksista: sovellukseen ei voi tehdä kirjanmerkkiä, siihen ei voi lähettää linkkiä sähköpostitse, eikä sitä voi twiitata. Hänen mukaansa mobiililaitteisiin tulisi rakentaa web-sovelluksia ja tekniikat niiden tekemiseen paranevat jatkuvasti.

Mikkonen ja Taivalsaari (2011; 2013; 2015) ovat käsitelleet natiivisovellusten ja avoimien web-palveluiden eroja useissa artikkeleissaan. Aikaisemmassa artikkelissaan (2011) he esittävät, että käynnissä on web- ja natiivitekniikoiden kilpailu, josta pidemmällä aikavälillä saattaa selvitä vain yksi voittaja. Myöhemmissä artikkeleissaan Mikkonen ja Taivalsaari ovat todenneet, että kehityssuunta näyttää eroavan mobiiliin ja perinteisten tietokone-ympäristöjen välillä: mobiililaitteissa natiivisovellukset ovat ottaneet merkittävän roolin, kun taas desktop-ympäristöissä kehityssuunta kulkee vahvasti kohti web-pohjaisten palveluiden dominointia.

Heitkötter et al. (2013) esittävät analyysinsä pohjalta, että hybridisovel-

lukset ovat suositeltu tapa toteuttaa sovelluksia usealle eri alustalle, ellei käyttöliittymän tarvitse muistuttaa tarkasti alustan natiivisovelluksia. Web-teknologiat sopivat heidän mukaansa hyvin alustojenvälisten sovellusten toteuttamiseen sillä ne ovat standardoituja, kohtuullisen yksinkertaisia, mutta tehokkaita ja hyvin tuettuja.

Aamulehto (2013) toteaa omassa työssään, että optimaalisen julkaisukeinon nimeäminen tietylle sovellustyyppille ei ole järkevää, vaan julkaisukeinon valinnassa pitäisi tapauskohtaisesti huomioida eri toteutusratkaisujen vahvuudet. Aamulehto kannustaa kuitenkin työnsä tulosten pohjalta web-tekniikoiden käyttöön. Hän toteaa, että modernit web-tekniikat ovat hyvin tuettuja kaikilla tutkituilla alustoilla ja tarjoavat hyvän julkaisualustan maailman levinneimpänä sovellusympäristönä.

Erkkilä (2013) puolestaan esittää, että natiivitekniikat tarjoavat parhaan käyttökokemuksen ja suorituskyvyn, mutta web-tekniikat tarjoavat kilpailukykyisen tavan tuottaa alustariippumattomia sovelluksia nopeasti ja joustavasti. Valintatilanteesta Erkkilä toteaa, että suositeltavin teknologia riippuu tapauskohtaisesti esimerkiksi sovelluksen kohdeyleisöstä, teknisistä vaatimuksista ja kaupallisista tavoitteista.

Ottka (2015) on muodostamaansa arviointimallia käyttäen arvioinut tutkimissaan kolmessa teollisuuskäyttöön tarkoitettussa sovelluksessa parhaaksi toteutustavaksi joko web-sivuston tai natiivisovelluksen. Eri toteutustapojen saamat pisteytykset olivat kuitenkin hyvin lähellä toisiaan ja yleispätevästi parasta toteutustapaa ei pysty niiden perusteella määrittelemään.

Ottka toteaa työn tulosten osoittavan, että alustariippumaton hybridiratkaisu voi olla mahdollinen, riippuen sovelluksen tarpeista. Hybridi-sovellukset voivat Ottkan mukaan olla hyväksyttävä vaihtoehto natiivisovelluksille jos toteutusbudjetti on tiukka, sovellus tarvitsee tehdä usealle mobiilialustalle tai tarvitsee pääsyn päätelaitteen ominaisuuksiin, joihin web-sovellus ei pääse. Web-sovelluksia hän suosittelee kevyisiin yrityssovelluksiin, jotka eivät vaadi pääsyä päätelaitteen kehittyneisiin ominaisuuksiin.

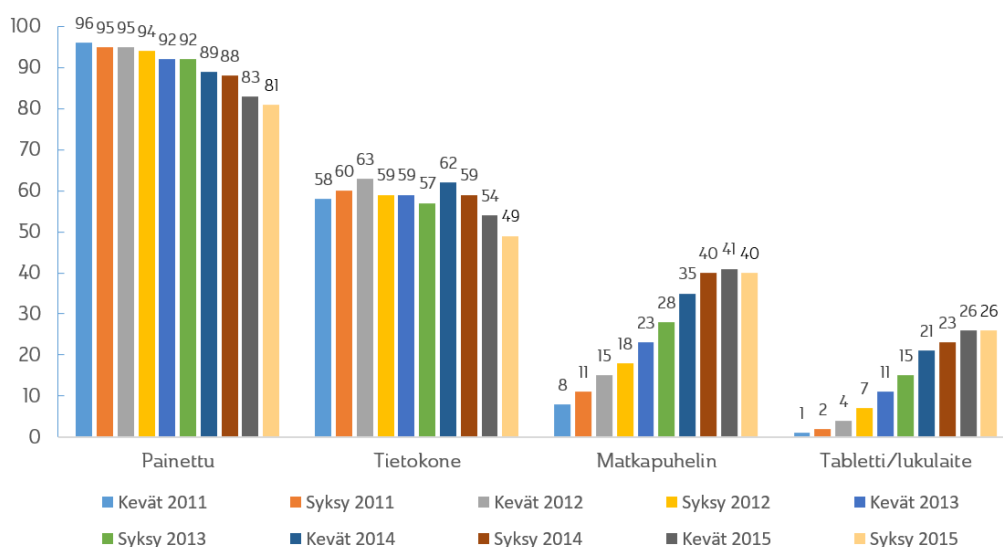
2.4 Käyttäjätilastot

Seuraavissa alakappaleissa käydään läpi muutamia tämän työn kannalta olennaisia tilastoja viime vuosilta. Kappaleessa 2.4.1 käydään läpi median käytössä tapahtuneita muutoksia. Kappaleessa 2.4.2 puolestaan käydään läpi tilastoja mobiilipalveluiden käytöstä.

2.4.1 Median käytön muutokset

Painettujen sanoma- ja aikakauslehtien lukeminen on ollut laskussa jo pitkään. Samaan aikaan digitaaliset kanavat ovat kasvattaneet suosiotaan. Erityisesti matkapuhelimien ja tablettien käyttäminen on kasvattanut suosiotaan. (Kansallinen Mediatutkimus, 2014)

Kuvassa 2.4 on esitetty Kansallisen mediatutkimuksen tulokset sanoma- ja aikakauslehtien lukemisesta eri lukuvälineillä vuosina 2011-2015. Luvut kuvaavat montako prosenttia vastaajista käyttää eri lukuvälineitä viikoittain.



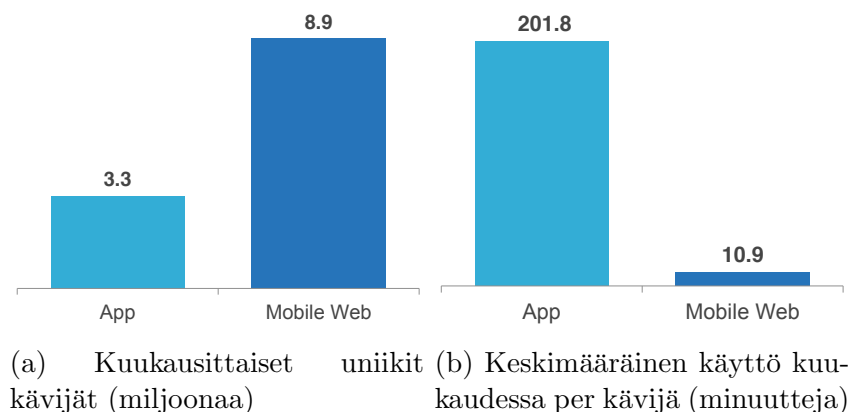
Kuva 2.4: Sanoma- ja aikakauslehtien lukeminen eri lukuvälineillä 2011-2015 (Kansallinen Mediatutkimus, 2015)

Kuvasta nähdään, että matkapuhelimien käyttö lehtien lukemiseen on tarkastelujakson aikana viisinkertaistunut 40 prosenttiin ja tablettien käyttö kasvanut yhdestä 26 prosenttiin. Samaan aikaan tietokoneen käyttö lukuvälineenä on pysynyt melko tasaisena. Painettu lehti on kuitenkin tutkimuksen mukaan säilyttänyt edelleen asemansa laajimmin käytettynä lukuvälineenä.

2.4.2 Mobiilipalveluiden käyttö

Comscore-yhtiön Yhdysvalloista keräämiin analytiikkoihin pohjautuva raportti (Lella et al., 2015) osoittaa, että mobiilisovellusten ja -sivustojen välillä on iso ero käyttötavoissa. Kuvassa 2.5 on esitetty sovellusten ja sivustojen kuukausittaiset uniikit kävijät ja keskimääräinen käyttö per kävijä.

Kuvasta nähdään, että sivustoilla on huomattavasti enemmän uniikkeja kävijöitä, mutta sovelluksia käytetään kävijää kohden huomattavasti enemmän kuin sivustoja. (Lella et al., 2015) esittävät, että sovelluksissa on haastavampaa kerätä laaja käyttäjäkunta, mutta saadut käyttäjät ovat lojaaleja palvelulle.



Kuva 2.5: Yhdysvaltojen 1000 suosituimmasta mobiilisovelluksesta ja mobiilisivustosta kerättyjen kävijätilastojen vertailu (Lella et al., 2015)

2.5 Yhteenveto aikaisemmista tutkimuksista

Mobiilipalveluiden jakeluun ja kehittämiseen on aikaisemmissa tutkimuksissa tutkittu kolmea lähestymistapaa: mobiilialustojen ohjelmistokauppojen kautta jaettavia natiivisovelluksia, web-selaimissa käytettäviä web-sivustoja sekä näitä yhdisteleviä hybridi-teknologioita.

Mobiilipalveluiden teknologiastrategiasta ja teknologiavalinnasta todetaan aikaisemmissa tutkimuksissa laajasti, että valinta riippuu tapauksesta ja tapauskohtaisesta tarpeista. Kullakin toteutustavalla on omat hyvät ja huonot puolensa, joita tulee painottaa tapauskohtaisesti. Yleispäteviä ohjeistuksia ei aikaisemmissa tutkimuksissa anneta.

Mobiilin uutispalvelun tarpeita tai teknologiavalintaa käsitteleviä tutkimuksia ei löytynyt, joten tässä diplomityössä tarjoutuu mahdollisuus kartoittaa uutispalvelun tarpeita mobiiliteknologioille, kuinka eri teknologiat tarpeisiin vastaavat ja periaatteita teknologiavalintaan uutispalvelun tapauksessa.

Luku 3

Case: Ilta-Sanomien mobiilit uutispalvelut

Tässä luvussa esitellään tämän työn tapaustutkimus, Ilta-Sanomien mobiilit uutispalvelut. Aluksi esitellään Ilta-Sanomat mediatoimijana ja organisaationa. Seuraavaksi esitellään Ilta-Sanomien kaikki mobiilipalvelut ja tarkemmin mobiilit uutispalvelut.

3.1 Ilta-Sanomat

Ilta-Sanomat on Suomen suosituin uutispalvelu. Ilta-Sanomilla on Kansallisen mediatutkimuksen (2015) mukaan eri kanavissa yhteensä 2 366 000 lukijaa viikossa. Ilta-Sanomat on osa Sanoma-konsernia ja vuoden 2014 alusta lähtien kuulunut Sanoma Media Finland -organisaatioon, johon on koottu Sanoma-konsernin Suomessa toimivat mediabrändit.

Ilta-Sanomat julkaisee painettua sanomalehteä kuutena päivänä viikossa, Urheilusanomat-lehteä viikoittain sekä erilaisia teemalehtiä noin kerran kuukaudessa. Digitaalisina palveluina Ilta-Sanomat tuottaa Iltasanomat.fi-uutispalvelun, Taloussanomat-uutispalvelun, Supersää-sääpalvelun sekä painettujen lehtien näköisversioita.

Mobiililaitteille Ilta-Sanomat tuottaa mobiilikäyttöön optimoituja versioita digitaalisista palveluistaan, sekä muutamia erityisesti mobiilikäyttöön suunniteltuja palveluita. Ilta-Sanomien mobiilipalvelut esitellään tarkemmin kappaleessa 3.2.

Painotuotteiden osalta Ilta-Sanomien tuotot tulevat sekä irtonumeromak-

suista että mainoksista. Digitaaliset palvelut ovat sen sijaan pääosin mainosrahoitteisia, pois lukien IS Extra -palvelu¹, josta peritään kuukausimaksua. IS Extra sisältää urheilutapahtumien live-videointeja, urheilun ja vedonlyönnin erikoisaineistoa sekä painettujen lehtien näköisversiot.

Painotuotteiden laskevan levikin ja digitaalisten palveluiden käytön kasvamisen myötä mainosrahoitteisten palveluiden ja mainosmyynnin merkitys on kasvanut. Ilta-Sanomilla digitaalinen mainonta on onnistunut kattamaan painotuotteiden puolella tapahtunutta laskua.

Ilta-Sanomissa online- ja mobiilimainonnan kasvu ylitti printtimainonnan ja levikin laskun. Online- ja mobiilimainonnan osuus on yli puolet Ilta-Sanomien mainosmyynnistä.

- Sanoman vuositulo 2013²

Digitaalisten palveluiden merkitys Ilta-Sanomille on jatkuvasti kasvussa. Kansallisen mediatutkimuksen (2015) mukaan Ilta-Sanomien lukijoista jo noin 80 prosenttia, eli noin 1.9 miljoonaa, käyttää ainostaan lehden digitaalisia kanavia. Lopuista noin 11 prosenttia lukee sekä digitaalisia kanavia että paperilehtiä ja noin 10 prosenttia pelkkiä paperilehteä. Digitaaliset palvelut tavoittavat siis moninkertaisen määrän lukijoita paperilehtiin verrattuna ja niiden osuus Ilta-Sanomien kokonaistavoittavuudessa on hyvin merkittävä.

3.2 Ilta-Sanomien mobiilipalvelut

Ilta-Sanomien tuottaa useita eri mobiilipalveluita: Ilta-Sanomien ja Taloussanomien uutispalveluita, IS Urheilu uutispalvelua, Supersää-palvelua sekä näköislehti-sovelluksia. Palveluiden kuvaukset ja tuetut alustat on esitelty taulukossa 3.1 ja kuvankaappaukset palveluista on kuvassa 3.1

Ilta-Sanomien ja Taloussanomien uutispalveluista tuotetaan mobiilipalveluita natiivisovelluksina eri alustoille sekä mobiililaitteille optimoituina web-sivustoina. Lisäksi Ilta-Sanomien on julkaissut erillisen IS Urheilu-sovelluksen, joka näyttää vain urheilu-uutisia sekä urheilutuloksia.

Uutissovellusten ohella Ilta-Sanomilla on erilliset näköislehti-sovellukset, joissa on ostettavissa digitaalinen versio päivän painetusta lehdestä.

¹<http://www.iltasanomat.fi/extra/>

²https://sanoma.com/sites/default/files/reports/sanoma_vuositulo_2013.pdf

Palvelu	Kuvaus	Alustat
Ilta-Sanomat	Uutispalvelu	Android, iOS, Windows Phone, Windows 8, mobiili web, responsiivinen web-sivusto
Taloussanomat	Talouteen keskittyvä uutispalvelu	Android, iOS, Windows Phone, mobiili web
IS Urheilu	Urheilu-uutispalvelu ja tulospalvelu	Windows Phone
IS Päivän lehti	Näköisversiot painetuista lehdistä	Android, iOS, Windows Phone
Supersää	Sääpalvelu	Android, iOS, Windows Phone, responsiivinen web-sivusto

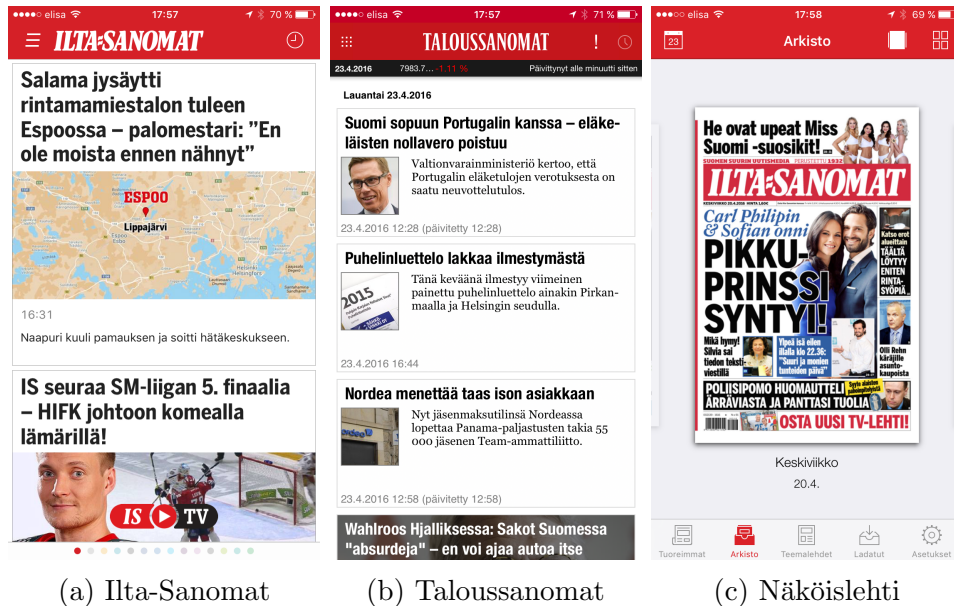
Taulukko 3.1: Ilta-Sanomien mobiilipalvelut

Supersää puolestaan on Ilta-Sanomien sääpalvelu, joka tehtiin *mobile first* -lähestymistavalla ensin mobiililaitteisiin. Palvelusta julkaistiin ensin natiivisovellus iOS-alustalle ja myöhemmin Windows Phone sekä Android -alustoille. Vasta näiden jälkeen Supersäästä julkaistiin responsiivinen web-sivusto.

Eri palveluiden osalta pyritään niiden saatavuuteen mahdollisimman monella mobiililaitteella. Suurin osa palveluista on saatavilla natiivisovelluksina kaikille kolmelle suosituimmalle mobiilialustalle Androidille, iOS:lle ja Windows Phonelle. Lisäksi suurin osa palveluista on saatavilla mobiililaitteille optimoituna web-sivustona.

Tässä tutkimuksessa keskitytään Ilta-Sanomien mobiiliin uutispalveluun, joka vastaa sisällöltään www.iltasanomat.fi -palvelua. Ilta-Sanomien uutispalvelun natiivisovelluksia on esitelty tarkemmin kappaleessa 3.3.1 ja mobiilisivustoa kappaleessa 3.3.2.

Ilta-Sanomien mobiilipalvelut ovat Suomen suosituimpia mobiilipalveluita. Mobiilipalveluiden käyttö on kasvanut merkittävästi viime vuosien aikana ja niistä tulee jo noin 60 prosenttia Ilta-Sanomien digitaalisten palveluiden käytöstä. Pääosa vierailumäärien vuosittaisesta 20–30 prosentin kasvusta on tullut mobiililaitteista. (Varjus, 2015)



(a) Ilta-Sanomat

(b) Taloussanomat

(c) Näköislehti



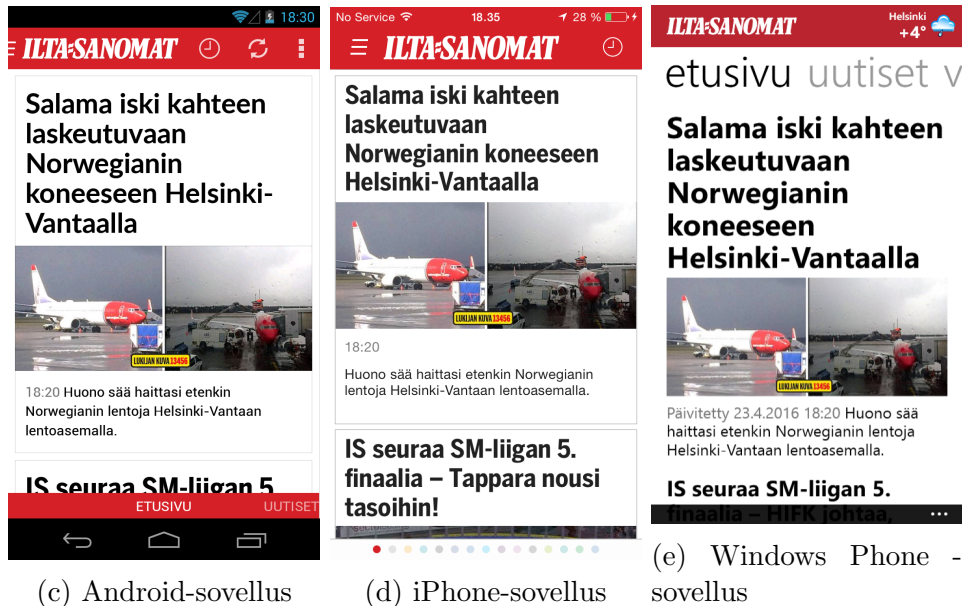
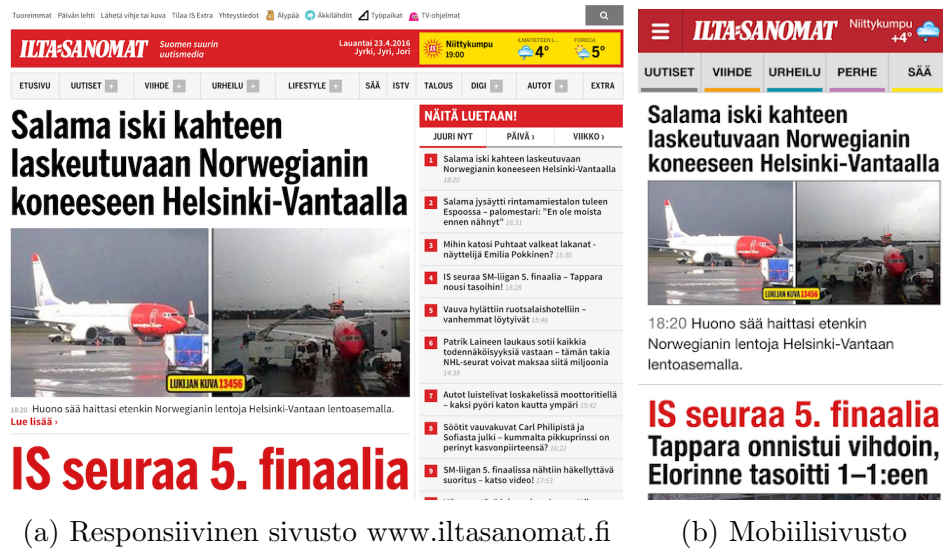
(d) Supersää

(e) IS Urheilu

Kuva 3.1: Ilta-Sanomien mobiilipalveluita

3.3 Ilta-Sanomien uutispalvelu

Uutispalvelu on Ilta-Sanomien mobiilipalveluista laajin ja käyttäjämääriltään ylivoimaisesti suurin. Sitä julkaistaan kuudessa eri kanavassa: Responsiivisena web-sivustona, mobiilisivustona sekä neljän alustan natiivisovelluksena. Kuvankaappaukset uutispalvelun etusivusta eri kanavissa on kuvassa 3.2.



Kuva 3.2: Ilta-Sanomien uutispalvelun etusivu eri kanavissa

3.3.1 Natiivit uutissovellukset

Ilta-Sanomien uutispalvelu on tällä hetkellä saatavissa natiivisovelluksena neljälle eri alustalle: iOS, Android, Windows Phone ja Windows 8.

Sovellukset on toteutettu käyttäen kunkin alustan natiiveja ohjelmistokehitystekniikoita. Sovellukset käyttävät yhteistä rajapintaa, mutta niiden välillä ei ole lainkaan yhteistä koodipohjaa.

Ilta-Sanomien sovellukset on suunniteltu erikseen kullekin mobiilialustalle ja ne pyrkivät noudattamaan kunkin alustan käyttöliittymäkonventioita.

3.3.2 Mobiilisivusto

Ilta-Sanomien mobiilisivusto *m.iltasanomat.fi* on mobiililaitteille optimoitu web-sivusto, joka tarjoaa suurimman osan *www.iltasanomat.fi* -sivuston sisällöstä.

Mobiilisivusto on suunniteltu palvelemaan kaikkia mobiilikäyttäjiä. Sivusto on toteutettu siten, että se toimii myös vanhemmilla ja vähemmän kehittyneillä mobiililaitteilla, kuten Nokian Symbian ja Asha -puhelimilla. Sivusto toimii myös ilman JavaScript-tukea.

Samaan aikaan uudempien älypuhelimien kykyjä ja ominaisuuksia on pyritty hyödyntämään ja tarjoamaan mahdollisimman hyvä palvelu kehittyneemmille laitteille. Esimerkiksi käyttäjän sijaintia hyödynnetään laitteilla jotka sitä tukevat ja korkean tarkkuuden näytöillä varustetuille laitteille tarjotaan korkealaatuisempia kuvia.

3.3.3 Responsiivinen web-sivusto

Ilta-Sanomien pääsivustosta *www.iltasanomat.fi* julkaistiin uusi versio keväällä 2015. Tämä uusi versio on *responsiivinen*, eli sen ulkoasu mukautuu käyttäjän päätelaitteen mukaan. Sivusto on suunniteltu tukemaan moderneja mobiili- ja tablet-laitteita. Käytännössä sivusto tukee Android, iOS ja Windows Phone -alustojen yleisimpiä versioita ja siten suurinta osaa mobiililaitteista. Vanhemmat laitteet ohjataan edelleen erilliselle mobiilisivustolle.

Responsiivista sivustoa ei ollut vielä julkaistu tämän työn materiaalin keräysvaiheen ja tulosten analysointiprosessin aikana. Työn tulokset perustuvat kappaleessa 3.3.2 esittelystä erillisestä mobiilisivustosta saatuihin kokemuksiin. Työn tulosten muodostamisen aikana *www.iltasanomat.fi* sivusto palveli ainoastaan desktop- ja tablet-käyttäjiä.

Luku 4

Materiaalit ja metodit

Tässä luvussa esitellään tutkimuksessa käytetyt materiaalit ja metodit. Ensin käydään läpi työn yleistä tutkimusmetodia. Seuraavaksi esitellään materiaalina käytetyt haastattelut ja prototyyppi-sovellukset.

4.1 Tutkimusmetodit

Tässä työssä tehty tutkimus on tyypiltään yhden tapauksen tapaustutkimus (engl. *single-case study*), jota tutkitaan reaalikontekstissa (Yin, 2003). Tapaustutkimuksen kohde, Ilta-Sanomien mobiilit uutispalvelut, on esitelty kappaleessa 3.

Tapaustutkimuksen pohjaksi tehtiin kirjallisuuskatsaus aikaisemmasta aihepiiriin tutkimuksesta. Tutkittavasta tapauksesta puolestaan kerättiin materiaali käyttämällä kolmea eri aineistoa:

1. Ilta-Sanomien mobiilipalveluiden parissa työskennelleiden *haastattelut*
2. Ilta-Sanomien uutissovelluksista tehdyt hybridi-*prototyypit*
3. *Osallistuva havainnointi* (engl. *participant observation*) osana Ilta-Sanomien mobiilipalveluita toteuttavaa tiimiä (Yin, 2003)

Kolmea eri aineistoa käyttämällä pyrittiin keräämään mahdollisimman kattava materiaali tapaustutkimuksesta. Usean aineistolähteen käyttäminen on Yinin (2003) mukaan olennainen periaate tapaustutkimuksen aineiston keräämisessä ja mahdollistaa aineistojen yhdistämisen *triangulaatiolla*.

4.2 Haastattelut

Tutkimuksen pääaineiston muodostaa Ilta-Sanomien mobiilipalveluiden parissa työskennelleiden henkilöiden haastattelut. Haastateltavat valittiin siten, että he edustavat kattavasti eri näkökulmia sekä Ilta-Sanomien omasta organisaatiosta, että teknistä kehitystä ja suunnittelua tekevän toimittajan, Futuricen puolelta. Tutkimusta varten tehtiin yhteensä 10 haastattelua. Haastatellut henkilöt on esitelty taulukossa 4.1.

Haastattelut järjestettiin kasvotusten, yksi haastateltava kerrallaan ja ne kestivät tunnista kahteen tuntiin. Haastattelut suoritettiin joulukuussa 2013 ja tammikuussa 2014. Haastateltavien luvalla kaikki haastattelut äänitettiin myöhempää analyysiä varten. Kaikki haastattelut pidettiin suomen kielellä, joten tarvetta kerätyn aineiston käänöksille ei ollut.

Haastattelut suoritettiin avoimina *puolistrukturoituina* haastatteluina. Haastatteluihin oli valmisteltu sekä kaikille yhteisiä, että haastateltavan rooliin mukaisia teemoja ja kysymyksiä. Haastatteluiden tavoitteena oli kerätä laaja-alaisesti haastateltavien näkemyksiä aihepiiristä, joten niiden muoto oli keskustelevalle ja kysymykset mahdollisimman avoimia. Haastattelukysymykset ovat liitteessä A.

Organisaatio	Rooli
Ilta-Sanomat	Mobiilipalveluiden tuottaja
	Online-kehityksen päällikkö
	Online-palveluiden johtaja
Sanoma-konserni	Mobiili- ja laitestrategian johtaja
Futurice	Käyttökokemussuunnittelija
	Asiakasvastaava
	Android-kehittäjä
	iOS-kehittäjä
	Windows Phone -kehittäjä
	Web-kehittäjä

Taulukko 4.1: Haastatellut henkilöt

4.2.1 Analyysimetodi: Grounded theory

Kerätty haastattelumateriaali analysoitiin käyttäen *grounded theory* -metodia. Grounded theory on aineistolähtöinen metodi, jonka esittelivät alunperin Glaser ja Strauss (1967). Siinä pyritään vaiheittain tunnistamaan teoria tutkimuksen aineiston pohjalta, sen sijaan että lähdettäisiin liikkeelle aikaisempien tutkimuksien teorioista. Metodilla on kehittäjiensä mukaan hyvä löytää täysin uusia teorioita. Grounded theory -metodi valittiin tähän työhön siksi, että tutkimuksen aihepiiristä ei ole olemassa kattavasti aikaisempia teorioita, joiden pohjalta materiaalia voisi analysoida.

Grounded theoryä sovellettiin tutkimuksessa Straussin ja Corbinin (1990) esittämien analyysimallien mukaisesti. Straussin ja Corbinin lähestymistapa on käytännönläheisempi kuin grounded theoryn toisen alkuperäisen kehittäjän, Glaserin. Straus ja Corbin esittävät valmiita työkaluja analyysin tekemiseen ja sallivat myös ulkopuolisen tiedon hyödyntämisen analyysin osana. (Warburton, 2006)

Aineistoa analysoitiin tutkimuskysymyksittäin Straussin ja Corbinin (1990) esittämien kolmen analyysivaiheen mukaisesti: *avoin koodaus*, *aksiaalinen koodaus* ja *selektiivinen koodaus*. Vaiheet käydään tarkemmin läpi seuraavassa alakappaleessa. Teorian mukaisesti näitä vaiheita tehtiin lomittain ja iteroiden kunnes materiaalista ei enää noussut esiin uusia asioita, eli tulokset olivat saturoituneet.

4.2.2 Haastattelujen analyysiprosessi

Haastattelumateriaalin analyysiprosessi muodostui useasta vaiheesta. Analyysiprosessin kaikki vaiheet, niiden tavoitteet ja lopputulokset on esitetty tiivistetysti taulukossa 4.2.

Haastattelujen analyysi aloitettiin litteroimalla äänitetyt haastattelut tarakoiksi tekstimuotoisiksi muistiinpanoiksi analyysiä varten. Haastatteluiden yhteiskesto oli noin 13 tuntia ja niistä kirjoitettujen muistiinpanojen määrä yhteensä 77 sivua.

Seuraavaksi muistiinpanot ryhmiteltiin haastattelu- ja tutkimuskysymyksittäin. Kunkin kolmen tutkimuskysymyksen alle ryhmiteltiin siihen liittyvät haastattelukysymykset ja niiden vastaukset kaikilta haastatelluilta. Tämän vaiheen tarkoituksena oli paremmin hallita muistiinpanojen isoa määrää ja tuottaa loogisia analysoitavia kokonaisuuksia. Grounded theoryn iterointiperiaatteen mukaisesti jaottelua myös uudelleenarvioitiin analyysiprosessin edetessä ja materiaalia siirrettiin tutkimuskysymysten välillä.

Seuraavaksi aloitettiin Straussin ja Corbinin (1990) analyysiprosessin mukainen *avoin koodaus*, jossa haastatteluvastauksia tulkittiin ja tehdyt havainnot käsitteellistettiin muodostaen *konsepteja*. Havaintoja vertailtiin keskenään ja samankaltaiset havainnot ryhmiteltiin samoihin konsepteihin.

Koodauksessa käytettiin apuna muun muassa *Post-it* lappuja. Kuvassa 4.1b näkyy kuinka kolmanteen tutkimuskysymykseen liittyviä konsepteja on kirjoitettu lapuille osana avointa koodausta. Eri värisiä lappuja hyödynnettiin myös erottelemaan positiivisia ja negatiivisia havaintoja.

Aksiaalisessa koodauksessa puolestaan luotiin yhteyksiä avoimessa koodauksessa muodostettujen konseptien välille ja muodostettiin niistä *kategorioita*. Aksiaalisen koodauksen tuloksena löydettiin viidestä kuuteen kategoriaa joista tutkimuskysymystä kohden. Näistä kategorioista muodostui lopulta tutkimustulokset tutkimuskysymyksiin.

Vaihe	Tavoite	Lopputulos
Litterointi	Haastattelumateriaalin muuttaminen tekstimuotoon analyysia varten.	Kirjoitetut muistiinpanot, yhteensä 77 sivua.
Ryhmittely	Haastattelumuistiinpanojen jaottelu analysoitaviksi kokonaisuuksiksi.	Muistiinpanot ryhmiteltyinä tutkimus- ja haastattelukysymyksittäin.
Avoin koodaus	Muistiinpanojen käsitteellistäminen, havaintojen ryhmittely konsepteiksi.	Konseptit, jotka kuvaavat materiaalista havaittuja seikkoja.
Aksiaalinen koodaus	Kategorioiden muodostaminen luomalla yhteyksiä konseptien välille.	Tutkimustulokset tutkimuskysymyksittäin. Löydettiin 5-6 kategoriaa per tutkimuskysymys.
Selektiivinen koodaus	Kategorioita peilataan toisiinsa ja pyritään löytämään niitä yhdistävä ydinkategoria.	Johtopäätökset, mobiiliteknologiastrategian viisi periaatetta. Yhtä yhtenäistä ydinkategoriaa ei löytynyt.

Taulukko 4.2: Haastattelumateriaalin analyysiprosessi



(a) Aksiaalinen koodaus, TK2

(b) Avoin koodaus, TK3

Kuva 4.1: Tulosten analysointia Post-it lappujen avulla

Kuvassa 4.1a on käynnissä aksiaalinen koodaus toisen tutkimuskysymyksen osalta. Konseptien välille on löydetty yhteyksiä ja niistä on muodostettu kategorioita vaakariveille. Kuvan oikeassa alakulmassa on ryhmittelemättömiä konsepteja, joille ei ole vielä löydetty kategoriaa.

Selektiivisessä koodauksessa kategorioita peilattiin toisiinsa ja pyrittiin Straussin ja Corbinin (1990) ohjeistuksen mukaisesti löytämään niitä yhdistävä ydinkategoria. Yhtä yhtenäistä ydinkategoriaa ei kuitenkaan löytynyt, sillä kategoriat olivat keskenään hyvin erityyppisiä. Tähän vaikutti luultavammin tutkimuskysymysten laajuus ja aihepiirin moniulotteisuus.

Ydinkategorian sijaan selektiivisen koodauksen prosessissa muodostui *mobileknologiastrategian* viisi periaatetta. Nämä periaatteet yhdistävät eri tutkimuskysymysten tuloksia ja pelaavat niitä keskenään muodostaen työn johdopäätökset.

4.3 Hybridi-prototyypit

Hybridi-lähestymistavan arviointia varten Ilta-Sanomat toteutti uutissovelluksesta prototyyppi-versiot, joissa hyödynnetään web-näkymiä osaan sisällön näyttämisestä. Prototyyppi-sovellusten toteutuksesta saatuja kokemuksia kerättiin tutkimuksen materiaaliksi osana haastatteluita.

Prototyyppi-sovellukset toteutettiin kolmelle mobiilialustalle: iOS, Android ja Windows Phone. Prototyypit toteutettiin kappaleessa 3.3.1 esiteltyjen nykyisten natiivien uutissovellusten pohjalta.

Prototyyppien toteutuksesta vastasi Ilta-Sanomilla ennastaan työskentelevä Futuricen tiimi. Alustakohtaisten prototyyppi-sovellusten toteutuksen teki kunkin alustan parissa työskennellyt kehittäjä ja taustajärjestelmän toteutuksen teki tämän diplomityön kirjoittaja.

Hybridi-sovellukseen web-sisältöjä palvelevan taustajärjestelmän toteutuksessa hyödynnettiin olemassaolevaa mobiilisivustoa. Mobiilisivustosta muokattiin versio, joka tarjoilee pelkän HTML-muotoisen sisällön ilman hybridi-sovelluksille turhia sivuston käyttöliittymän elementtejä, kuten valikot ja sivun alaosan *footer*-osio.

Hybridi-prototyypeissa haluttiin ensisijaisesti testata palvelun etusivun soveltuvuutta hybridi-lähestymistapaan. Etusivun lisäksi web-näkyminä toteutettiin osiosivut, eli esimerkiksi urheilu ja viihde -osiot. Siirtymät näkymien välillä toteutettiin natiivisti. Artikkelinäköymät jätettiin natiivitoteutukseksi. Natiivikoodissa käsiteltiin web-näkymästä tulleet interaktio-tapahtumat. Esimerkiksi artikkelin linkin painaminen web-näkymässä tulkittiin vastaavaksi natiivi-interaktioksi, joka avasi natiivin artikkelinäköymän.

Kuvassa 4.2 on kuvankaappaukset Ilta-Sanomien etusivusta iOS-alustalla hybridi-prototyypissä ja mobiilisivustolla. Kuvasta nähdään, että etusivun sisältöosa on identtinen prototyypissä ja mobiilisivustolla. Hybridi-prototyypin etusivun sisältö on web-näkymään piirrettyä HTML-sisältöä. Valikon, logon ja säätilan sisältävä näkymän yläosa on hybridi-prototyypissä edelleen natiivi, samoin kuin näkymän alareunassa oleva sivuttain vieritettävien näkymien indikaattori.



(a) Hybridi-prototyyppi



(b) Mobiilisivusto

Kuva 4.2: Ilta-Sanomien etusivu hybridi-prototyypissä ja mobiilisivustolla

Luku 5

Tulokset

Tässä luvussa esitellään tutkimuksen tulokset tutkimuskysymyksittäin. Ensin käydään läpi minkälaisia vaatimuksia uutispalvelulla on käytettävälle mobiiliteknologioille. Seuraavaksi tutkitaan nykyisiä web- ja natiivisovelluksia uutispalvelun jakelukanavina. Lopuksi tutkitaan voisiko hybridi-sovelluksella web- ja natiivitekniikoita yhdistelemällä vastata entistä paremmin uutispalvelun vaatimuksiin.

5.1 Uutispalvelun vaatimukset mobiiliteknologioille

Tässä kappaleessa käydään läpi tulokset ensimmäiseen tutkimuskysymykseen: *Minkälaisia vaatimuksia uutispalvelulla on käytettävälle mobiiliteknologioille?*

Tutkimuksessa kerätyistä materiaaleista identifioitiin viisi päävaatimusta sekä yksi tulevaisuudessa todennäköinen vaatimus:

1. Sisällön yhteneväisyys, toimivuus ja hallittavuus kaikissa kanavissa
2. Muokattavuus: nopeat ja hallitut muutokset kaikkiin kanaviin
3. Käyttökokemus ja ulkoasu: tuntuu brändin mukaiselta kaikissa kanavissa, mutta noudattaa alustojen konventioita
4. Kaupalliset tavoitteet: mainosinventaarin kasvattaminen laajalla käyttäjäpohjalla, säännöllisellä käytöllä ja uusilla mainostyypeillä

5. Nopeus ja luotettavuus: sisällön julkaiseminen on nopeaa, palvelu toimii nopeasti ja luotettavasti kaikissa tilanteissa
6. Personointi: tulevaisuuden vaatimus

Seuraavissa alakappaleissa käydään löydetty vaatimukset yksitellen läpi ja lopuksi tehdään yhteenveto vaatimuksista.

5.1.1 Sisällön yhteneväisyys, toimivuus ja hallittavuus

Ilta-Sanomien mobiilipalvelusta vastaavien henkilöiden haastatteluissa toistui tärkeimpänä vaatimuksena sisällön toimivuus kaikissa kanavissa yhteneväisenä ja hallittavana kokonaisuutena. Sisältö on uutispalvelun määrittävä osa, jonka ympärille koko palvelu on rakennettu.

Kaikki uutistoimituksen tuottama sisältö tulisi olla saatavilla kaikille käyttäjille, riippumatta siitä käyttävätkö he palvelua pöytäkoneella tai mobiililaitteella. Eri laitteiden välillä ei myöskään saisi olla eroja näytettävän sisällön määrässä tai toimivuudessa.

Minulla on tavoite, että mobiilissa kaikki eri tavat joilla käyttäjä voi meidän sisältöä käyttää olisivat *sisällöltään ja ominaisuuksiltaan suurinpiirtein samalla viivalla*, jotta käyttäjällä ei pitäisi olla merkitystä sillä käyttääkö sovellusta vai sivustoa vai mitä ikinä käyttäekin. Käyttäjä voi luottaa että meidän sisältö on siellä ja toimii ja ominaisuudet ovat suurinpiirtein synkassa.

- Ilta-Sanomien mobiilipalveluiden tuottaja

Mobiilipalvelut hahmotetaan uutisorganisaatiossa yhtenä kanavana, jossa sisältö joko toimii tai ei toimi. Teknisesti kyseessä on paljon hienojakoisempi kokonaisuus, mutta varsinaisen kehitystiimin ulkopuolella mobiilipalveluita ei hahmoteta erillisiksi palveluiksi ja niiden eri versioiksi. Tästä johtuen eri mobiilipalvelut pitäisi teknisesti saada sisällöltään mahdollisimman yhtenäisiksi.

Toimituksen pitäisi pystyä hallitsemaan sisältöä siten, että he pystyvät luottamaan sisällön toimivan eri kanavissa halutulla ja ennakoitavalla tavalla. Kun toimituksessa julkaistaan sisältöä, ei heillä ole mahdollisuutta testata sen toimivuutta kaikilla eri päätelaitteilla. Kaiken sisällön pitää olla siirrettävissä kaikkiin kanaviin tarkoituksenmukaisessa muodossa.

Etusivu ja listaukset

Uutispalvelun tärkein näkymä on etusivu. Toimitus hallitsee manuaalisesti etusivulla näytettävää sisältöä ja nostaa siihen päivän tärkeitä uutisaiheita. Etusivu muodostaa toimituksen *kuratoiman* katsauksen sen hetken uutisaiheisiin. Etusivun ylläpitämiseen käytetään paljon vaivaa ja se olisikin suotavaa saada näkymään mahdollisimman samansisältöisenä kaikissa kanavissa.

Ilta-Sanomien etusivu sisältää eri uutiskategorioita, kuten pääuutiset, urheilu ja viihde. Lisäksi etusivulla on muiden saman konsernin uutispalveluiden uutisia. Näytettävät kategoriat, niiden järjestys ja kustakin kategoriasta näytettävien uutisten määrä tulisi olla yhteneväinen.

Artikkeleista luodaan etusivua varten erilliset *nostot*, jotka sisältävät mm. otsikon, ingressin, kuvan ja *stickerin* eli tekstin alussa olevan avainsanakomponentin. Nämä muodostavat keskenään kokonaisuuden, jonka toimitaja on koostanut. Tästä kokonaisuudesta ei yleensä voi ottaa mitään osaa pois, koska ne on suunniteltu toimimaan yhdessä. Nostoilla on eri ulkoasupohjia (engl. *template*), jotka määrittelevät miten noston sisältö näytetään: esimerkiksi kuvan ja otsikon koot sekä sijainnin toisiinsa nähden.

Etusivun nostot pitäisi saada näkymään kaikissa kanavissa siten, että niiden sisältö ja muodostama kokonaisuus välittyy samanlaisena joka kanavaan. Uutisten keskinäinen hierarkia on myös olennaista välittyä läpi kaikkiin kanaviin. Mobiilisovellusten tulisi noudattaa ulkoasu-pohjien mukaista asettelua, jotta nostot näkyisivät samanlaisina kaikissa kanavissa.

Myös muissa palvelun listauksissa, kuten osio-näkymissä, tulisi noudattaa yhtenäistä logiikkaa ja sisällön tulisi olla yhteneväistä, jotta käyttäjä pystyisi johdonmukaisesti löytämään saman sisällön kaikista kanavista.

Monimediainen sisältö

Uutispalvelun sisällön pitää olla nykyään entistä monipuolisempaa. Sisältö voi tekstin ja kuvien lisäksi koostua esimerkiksi videoista tai erilaisista artikkeleihin upotetuista komponenteista. Upotettuja komponentteja voivat olla palvelun omat tai kolmannen osapuolen komponentit. Palvelun omia komponentteja ovat esimerkiksi äänestys-, live seuranta- ja urheilutulos -komponentit. Kolmannen osapuolen komponentteja ovat esimerkiksi Twitter tai Instagram -palveluiden tarjoamat sisältökomponentit.

Sisällön eri komponentit muodostavat kokonaisuuden, joka tulisi esittää yhtenäisenä. Jos jokin osa siitä ei toimi, niin kokonaisuus jää vajaaksi ja huo-

noimmassa tapauksessa sen merkitys muuttuu. Tästä syystä on tärkeää että eri komponenttien toimivuus mobiililaitteissa pystyttäisiin varmistamaan. Mikäli kaikkia sisällön komponentteja ei kuitenkaan ole teknisesti mahdollista saada toimimaan kaikissa kanavissa, olisi huolehdittava että ne eivät ainakaan aiheuta ongelmia sovellukseen ja mielellään kerrottava käyttäjälle selkeästi mitä sisältöä puuttuu. Erityisesti videoiden rooli uutispalveluissa on noussut merkittäväksi, joten niiden toimivuus kaikissa kanavissa on tärkeää.

5.1.2 Muokattavuus: nopeat ja hallitut muutokset

Toisena merkittävänä vaatimuksena Ilta-Sanomien mobiilipalveluista vastaavien henkilöiden haastatteluissa nousi esiin nopeiden muutosten tekeminen palveluun siten, että kokonaisuus pysyy hallittavana.

Mobiilipalveluiden kehittämisen pitäisi olla mahdollisimman joustavaa ja tarvittaessa muutoksia pitäisi pystyä tekemään nopeasti siten, että ei vaaraneta edellisessä kappaleessa käsiteltyä vaatimusta sisällön yhteneväisyydestä ja hallittavuudesta.

Muutostarpeita saattaa tulla hyvin nopeasti. Uutismaailmassa toimitaan hyvin lyhyellä suunnittelujaksolla ja lisäksi äkillisiä uutistapahtumia tulee ja niihin pitää pystyä reagoimaan. Uutispuolen asioita suunnitellaan usein seuraavalle viikolle tai kahdelle, eikä pidemmän aikavälin suunnitelmia välttämättä ole mahdollista tehdä.

Pieniä muutoksia pitäisi pystyä viemään jopa päivässä eteenpäin kaikkiin kanaviin. Esimerkki tällaisesta muutoksesta voisi olla etusivulle sijoitettava interaktiivinen komponentti, joka liittyy ajankohtaiseen uutistapahtumaan. Desktop-sivuston puolella tällaisia muutoksia on totuttu tekemään ja samantyyppistä reagointikykyä edellytetään myös mobiilipalveluilta.

Toimituksen pitäisi pystyä oletamaan että uusi ominaisuus toimii luotettavasti kaikissa kanavissa. Tarvittaessa uudelle ominaisuudelle pitää tehdä erilaisia varmistuksia (engl. *fallback*), joilla vanhempien versioiden ja huonompien päätelaitteiden käyttäjille näytetään uuden ominaisuuden sijasta korvaava versio sisällöstä tai ilmoitus sisällön toimimattomuudesta. Varmistuksia pitäisi kuitenkin pyrkiä välttämään, koska ne aiheuttavat lisää kompleksisuutta kokonaiskuvaan.

5.1.3 Käyttökokemus ja ulkoasu: tuntuu brändin mukaiselta ja noudattaa alustojen konventioita

Kolmantena isona vaatimuksena ilmeni mobiilipalvelun ulkoasun ja käyttöliittymän näkyminen palvelun brändin mukaisena kaikissa kanavissa, mutta samalla huomioiden eri alustojen käyttöliittymäkonventiot ja tarjoten parhaan mahdollisen käyttökokemuksen.

Uutispalvelun brändin pitää olla tunnistettava kaikissa kanavissa. Sisällön ulkoasu viestii palvelun brändiä ja on myös osa journalistista ilmaisua. Sisällön ulkoasun pitäisi olla tarpeeksi yhdenmukainen kaikissa kanavissa, jotta toimittajat pystyvät ennakoimaan miten sisältö näkyy eri kanavissa ja asettamaan ulkoasun sen mukaisesti.

Iltapäivälehtien osalta palvelun ulkoasussa haetaan *iltapäivälehtimäisyyttä*, joka on osa palvelun brändiä. Erityisen tärkeää iltapäivälehtimäisyys on etusivun osalta. Iltapäivälehtimäisyys muodostuu useista erilaisista elementeistä, esimerkiksi fonteista, isoista otsikoista, isoista kuvista, väreistä ja näiden muodostamasta kokonaisuudesta, jonka hierarkian ja mielikuvan tulisi säilyä samana kaikissa kanavissa.

Käyttöliittymän tulisi kuitenkin tarjota paras mahdollinen käyttökokemus kullekin mobiilialustalle. Käyttäjä on tottunut käyttämään alustan palveluita tietyllä tavalla ja palvelun ei ole syytä rikkoa sitä käyttötapaa. Palvelun käyttö pitää olla mahdollisimman kitkatonta kullakin alustalla. Tämä edellyttää alustan käyttöliittymäkonventioiden noudattamista. Esimerkiksi navigaattiorakenteet olisi syytä olla alustan konventioiden mukaisia.

Uutispalvelun käyttökokemuksen ja ulkoasun suunnittelu eri alustoille vaatii sopivan kompromissin hakemista käyttöliittymien yhteneväisyyden ja alustakohtaisten konventioiden noudattamisen välille.

5.1.4 Kaupalliset tavoitteet: mainosinventaarin kasvattaminen laajalla käyttäjäpohjalla, säännöllisellä käytöllä ja uusilla mainostyypeillä

Kuluttajalle ilmaisen uutispalvelun tapauksessa palvelun tuotot tulevat mainosten näyttämisestä loppukäyttäjille. *Mainosinventaarin* kasvattaminen onkin kaupallisesta näkökulmasta mobiilipalveluiden tärkein tavoite. Mainosinventaarilla tarkoitetaan palvelun *mainosnäyttöjä*, joita voidaan myydä mainostajille. Mainosinventari koostuu mainosnäyttöjen määrästä sekä laadullisesta valikoimasta, eli eri tyyppisistä mainospaikoista eri palveluissa.

Jotta mainosnäyttöjä saataisiin mahdollisimman paljon, tulisi mobiilipalvelut toteuttaa siten, että ne ovat mahdollisimman laajan käyttäjäjoukon käytettävissä ja niitä käytetään mahdollisimman säännöllisesti sekä mahdollisimman paljon kerrallaan.

Uutispalvelun tulisi olla saatavilla jollain tavalla kaikille eri päätelaitteille, jotta se olisi mahdollisimman laajan käyttäjäjoukon käytettävissä. Osalla laitteista tuki voi olla huonompi, mutta kaikkien halukkaiden käyttäjien tulee jotenkin päästä näkemään palvelun sisältöä.

Mainosnäyttöjen määrän kasvattamisen kannalta on tärkeää saada palvelulle *aktiivisia käyttäjiä*, jotka käyttävät palvelua mahdollisimman säännöllisesti ja mahdollisimman paljon kerrallaan. Säännöllistä käyttöä voidaan edistää tuomalla jatkuvasti uutta mielenkiintoista sisältöä ja houkuttelemalla käyttäjiä uuden sisällön pariin esimerkiksi käyttäjälle lähetettävien *notifi-kaatioiden* avulla.

Käyttösession pituutta puolestaan voidaan edistää tekemällä palvelun käytöstä mahdollisimman vaivatonta ja nopeaa, sekä tarjoamalla mekanismeja joilla käyttäjä löytää helposti lisää mielenkiintoista sisältöä.

Uudet mainostyypit

Mainosinventaaaria tulisi laajentaa myös tarjoamalla erilaisia mainostyyppejä. Perinteisten kuvapohjaisten mainosten lisäksi mobiilikanaviin halutaan tarjota uusia mainostyyppejä, joista mainostajat ovat valmiita maksamaan enemmän. Tällaisia mainostyyppejä ovat esimerkiksi *interaktiiviset mainokset* ja *koko sivun mainokset*.

Interaktiivisissa mainoksissa käyttäjä pystyy esimerkiksi selaamaan mainoksen eri osia, katsomaan kartalta mainostajan toimipaikkoja tai tilaamaan tekstiviestinä kampanjaviestejä. Interaktiiviset mainokset perustuvat useimmiten web-tekniikoihin.

Koko sivun mainokset täyttävät päätelaitteen ruudun kokonaan ja sisältävät usein myös interaktiivisia komponentteja. Koko sivun mainokset ovat mainostajien silmissä erityisen houkuttelevia mainostyyppejä, koska mainostaja saa koko ruudun haltuunsa. Niistä ollaan valmiita maksamaan enemmän ja erityisesti tablettikoossa koko sivun mainokset ovatkin kaikkein tuottoisimpia mainostyyppejä.

Koko sivun mainokset sijoitetaan vaakasuunnassa pyyhkäistävien näkymien väliin, esimerkiksi käyttäjän selatessa uutisia pyyhkäisemällä. Tästä syystä sovelluksissa tulee suosia käyttöliittymäratkaisuja, jotka mahdollistavat koko

sivun mainosten luontevan sijoittamisen osaksi käyttökokemusta.

Uusienkin mainostyyppien pitäisi toimia kaikissa kanavissa oikein. Tämä on haaste erityisesti interaktiivisten mainosten osalta. Mainokset myydään usein mobiilikanavaan yhtenä kokonaisuutena, joten olisi tärkeää, että mainos toimisi yhteneväisesti kaikissa mobiilisovelluksissa ja kokonaisuus olisi hallittavissa.

Jos mainokset näkyvät tai toimivat oikein vain osassa sovelluksia, on kokonaisuuden hahmottaminen ja hallinta hankalaa sekä mainosmyyjille että mainoksia ostaville mainostajille. Tämä aiheuttaa merkittävästi ylimääräistä työtä, kun mainosten toimivuus eri sovelluksissa joudutaan erikseen testaamaan, mainoksista joudutaan tekemään useita erilaisia versioita ja eri sovelluksiin joudutaan *trafikoimaan* eri versiot mainoksesta.

5.1.5 Nopeus ja luotettavuus: sisällön julkaiseminen nopeaa, palvelu toimii nopeasti ja luotettavasti kaikissa tilanteissa

Viidentenä vaatimuksena haastatteluista nousi esiin nopeuden ja luotettavuuden tärkeys uutispalveluille. Uutispalveluille on tärkeää, että julkaistu sisältö on mahdollisimman nopeasti käyttäjien saatavilla, palveluiden käyttö on nopeaa ja palvelut toimivat kaikissa tilanteissa luotettavasti.

Vaatus sisällön päivittymisen nopeudesta on uutispalveluiden erityispiirre, joka korostuu verrattuna moniin muihin mobiilipalveluihin. Jokainen uutispalvelu haluaa olla ensimmäisten joukossa kertomassa ajankohtaisista uutisista. Kun toimitus julkaisee uutisen, pitää sen olla välittömästi saatavilla kaikissa julkaisukanavissa. Tällöin on tärkeää päästä eroon ylimääräisistä viiveistä niin julkaisuprosessissa, taustajärjestelmissä kuin loppukäyttäjän mobiilipalvelussakin.

Käyttäjille lähetettävät *notifikaatiot* auttavat uuden uutisen välittämässä käyttäjien tietoon mahdollisimman nopeasti. Mahdollisuus vastaanottaa notifikaatioita onkin tärkeä ominaisuus, jota uutispalvelut haluavat tarjota aktiivisille käyttäjilleen.

Myös mobiilipalvelun suorituskyky ja käytön nopeus on tärkeää, sillä mitä nopeammin palvelu käyttäjän laitteessa toimii, sitä enemmän palvelua käytetään. Mobiilipalveluiden tapauksessa käyttötilanteet ovat usein erityisen nopeita, jopa vain muutamia sekunteja, jolloin palvelun toimintanopeus korostuu entisestään. Nopeissa käyttötilanteissa kilpaillaan muiden palveluiden kanssa käyttäjän ajasta. Palvelun nopeus on tärkeää tekijä kun käyttäjä

valitsee mitä palvelua hän käyttää.

Mobiilin uutispalvelun nopeuden optimoimiseen tuovat oman lisähaasteensa huonot ja vaihtelevat verkkoyhteydet, joissa mobiilipalveluita usein käytetään. Uutispalveluita käytetään paljon esimerkiksi julkisissa liikennevälineissä, joissa verkkoyhteyden laatu vaihtelee matkan aikana. Esimerkiksi junamatkan aikana verkkoyhteys voi välillä puuttua kokonaan. Tällöin uutisten lukeminen ei luonnollisesti onnistu, elleivät ne ole valmiiksi käyttäjän laitteen muistissa.

Toimiva välimuistitus (engl. *caching*) sekä riittävä offline-tuki ovatkin erittäin hyödyllisiä mobiilille uutispalvelulle. Hyvä välimuistitratkaisu voi esimerkiksi ladata etukäteen laitteen muistiin ne artikkelit, jotka käyttäjä todennäköisesti lukee. Myös erilaisten virhetilanteiden hallinta on olennaista. Esimerkiksi sisällön lataaminen saattaa keskeytyä verkkoyhteyden heiketessä tai sisällöstä saattaa olla välimuistissa liian vanha versio.

Heikkojen verkkoyhteyksien tukemiseen sekä yleiseen palvelun nopeuttamiseen vaikuttaa myös palvelun siirtämän datamäärän optimoiminen mahdollisimman pieneksi. Minimoimalla siirrettävä datamäärä saadaan sisältö näkymään mahdollisimman nopeasti käyttäjälle ja vähennetään käyttäjän tiedonsiirtokustannuksia.

Uutispalvelun luotettavuuden ja toimintavarmuuden maksimointi on tärkeää. Käyttäjien pitää pystyä luottamaan siihen, että he saavat luettua uutiset silloin kun haluavat. Jo pienetkin ongelmat laskevat käyttäjien luottamusta ja saattavat katkaista lukijasuhteen. Mobiilipalveluiden luotettavuuden vaatimusta on kasvattanut myös mobiilipalveluiden suosion kasvu uutispalvelun jakelukanavana, kuten kappaleessa 3 käsiteltiin.

Luotettavuuden maksimoimiseksi uutispalvelun toteutusteknologiat tulisi valita siten, että ne takaavat mahdollisimman toimintavarman palvelun. Usein tämä tarkoittaa pidättäytymistä luotettaviksi todistetuissa teknologioissa sen sijaan, että otetaan etupainotteisesti käyttöön kaikkein uusimpia teknologioita. Uusimpien teknologioiden pelätään olevan keskeneräisiä ja aiheuttavan stabiiliusongelmia.

5.1.6 Personointi: tulevaisuuden vaatimus

Haastatellut Ilta-Sanomien edustajat toivovat, että palveluun voitaisiin tulevaisuudessa tuoda lisää henkilökohtaisuutta ja relevanttiutta käyttäjälle. Palvelua tulisi siis olla mahdollista *personoida* kunkin käyttäjän tarpeiden mukaiseksi.

Palvelussa näytettävää sisältöä voitaisiin mukauttaa käyttäjän käyttötottumusten mukaan. Etusivulle voitaisiin nostaa enemmän uutisia kategorioista, joista käyttäjä on kiinnostunut. Valinnat voisi tehdä joko automaattisesti käyttäjätietojen analysoinnin pohjalta, tai antaa käyttäjän itse valita häntä kiinnostavia uutistyyppisiä.

Personointiin liittyy myös palvelun *kontekstuaalisuuden* lisääminen. Automaattinen mukautus voisi ottaa huomioon myös käyttäjän sen hetkisen käyttökontekstin. Esimerkiksi käyttäjätietojen analysoinnin perusteella voitaisiin arvioida minkälaisista uutisista käyttäjä juuri siinä tilanteessa voisi olla kiinnostunut.

Toisaalta palvelu voidaan tehdä relevantimpi käyttäjälle myös tuomalla eri sisältöjä erillisinä palveluina tarkemmin rajatuille kohderyhmälle. Esimerkiksi urheilusta kiinnostuneille voidaan tehdä urheilu-uutissovellus, joka näyttää urheiluun liittyvää sisältöä. Tällä lähestymistavalla on myös mahdollista lisätä palveluiden henkilökohtaisuutta.

5.1.7 Yhteenveto

Tutkimuksessa identifioidut uutispalvelun tärkeimmät vaatimukset mobiiliteknologioille on esitelty taulukossa 5.1.

Vaatus	Kuvas
Sisällön yhteneväisyys, toimivuus ja hallittavuus	Sisällön tulee toimia kaikissa kanavissa yhteneväisenä ja helposti hallittavana kokonaisuutena.
Muokattavuus	Teknologioiden tulee mahdollistaa nopeat ja hallitut muutokset palvelun sisältöön.
Käyttökokemus ja ulkoasu	Palvelun käyttökokemuksen tulee tuntua uutispalvelun brändin mukaiselta ja noudattaa alustojen käyttöliittymäkonventioita.
Kaupalliset tavoitteet	Mainostuottojen takaaminen mainosinventaarilla kasvattamalla: laajalla käyttäjäpohjalla, säännöllisellä käytöllä ja uusilla mainostyypeillä.
Nopeus ja luotettavuus	Julkaistun sisällön tulee olla nopeasti käyttäjien saatavilla, palvelun käytön olla nopeaa ja palvelun toimia kaikissa tilanteissa luotettavasti.
Personointi	Tulevaisuuden vaatimus. Teknologioiden tulee jatkossa mahdollistaa palvelun sisällön personointi käyttäjäkohtaisesti tai kontekstuaalisesti.

Taulukko 5.1: Yhteenveto tutkimuksessa identifioituista uutispalvelun vaatimuksista

5.2 Miten web- ja natiiviteknologiat vastaavat uutispalvelun vaatimuksiin?

Tässä kappaleessa käsitellään tutkimuksen toista tutkimuskysymystä: *Miten web- ja natiiviteknologiat vastaavat uutispalvelun vaatimuksiin ja minkälaisia ongelmia niiden kanssa nykyisellään on?*

Tutkimuskysymystä tarkastellaan Ilta-Sanomien mobiilipalveluiden kautta. Tarkastelun kohteena olevat mobiilipalvelut esiteltiin luvussa 3. Ilta-Sanomien mobiilipalveluita peilataan kappaleessa 5.1 identifioituihin uutispalvelun vaatimuksiin mobiiliteknologioille ja arvioidaan miten ne nykyisellään täyttävät vaatimukset.

Aluksi käydään läpi natiivisovellusten ja mobiilin web-sivuston eroja uutispalvelun jakelukanavina sekä eroja niiden käyttötavoissa. Seuraavissa alakappaleissa käsitellään uutispalvelun tärkeimpien vaatimusten - sisällön yhteneväisyyden, muokattavuuden, käyttökokemuksen, kaupallisten tavoitteiden sekä nopeuden ja luotettavuuden - täyttymistä ja ongelmia nykyisissä mobiilikanavissa.

5.2.1 Natiivisovellusten ja web-sivuston erot jakelukanavina ja käyttötavoissa

Haastatteluissa kävi ilmi, että natiivisovellusten ja mobiililaitteille suunnatun web-sivuston nähdään vastaavan pääosin samoihin käyttötapauksiin. Käyttäjän henkilökohtaisista preferensseistä riippuu kumpaa kanavaa hän haluaa mobiilissa käyttää.

Mobiililaitteille optimoitu web-sivusto on uutispalvelun tapauksessa välttämätön edellytys, joka täytyy olla olemassa linkitettävyyden ja yleisen saatavuuden takia. Natiivisovellukset puolestaan tarjoavat lisätoimintoja ja edistävät säännöllisempää käyttöä, joka on kaupallisesti erittäin tärkeä asia. Molemmat kanavat nähdään tarpeellisina uutispalvelulle ja ne tukevat toisiaan.

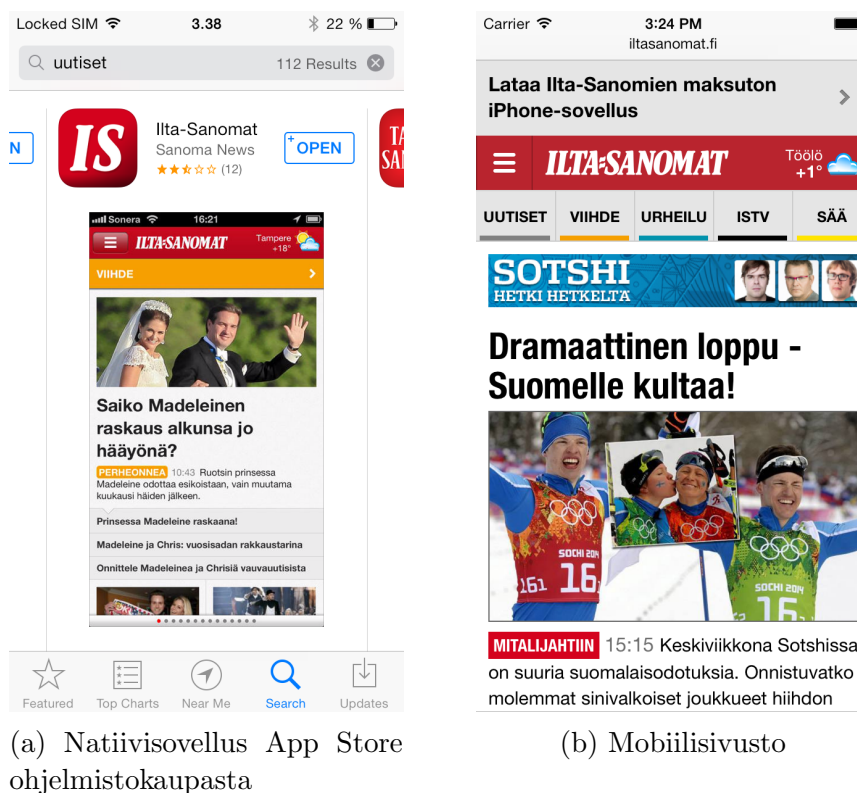
Erot mobiilipalvelun jakelukanavina

Sovellukset ja web-sivusto ovat palvelun jakelukanavina hyvin erilaisia. Natiivisovellukset pitää asentaa käyttäjän laitteeseen, kun taas web-sivustoa käytetään WWW-selaimella ilman erillistä asennusta.

Web-sivuston suurimpia etuja ovat *linkitettävyyys, jaettavuus* ja spontaanin

käytön mahdollistaminen. Sivustolle voi linkittää esimerkiksi sosiaalisen median palveluista ja hakukoneista. Myös muista saman konsernin palveluista voidaan tehdä *kävijäohjausta* linkittämällä kiinnostavaan sisältöön toisessa uutispalvelussa. Uutispalvelun tapauksessa linkitettävyyden on erittäin tärkeä edellytys mobiilipalvelulle, sillä erityisesti sosiaalista mediaa käytetään paljon mobiililaitteilla ja palvelun sisällön on oltava helposti saavutettavissa myös mobiilissa.

Mobiili web-sivusto nähdäänkin pakollisena edellytyksenä, joka uutispalvelulla pitää joka tapauksessa olla olemassa. Uutispalvelulla tulee olla URL-osoite, joka toimii kaikilla mobiililaitteilla, jotta kaikki potentiaaliset käyttäjät voivat halutessaan katsoa palvelun sisältöä.



Kuva 5.1: Ilta-Sanomien mobiilipalvelun jakelukanavat iPhone-laitteella

Natiivisovellus asennetaan alustan ohjelmistokaupasta, joka toimii samalla myös markkinointikanavana sovellukselle. Ohjelmistokaupan kautta käyttäjä saattaa löytää palvelun spontaanisti esimerkiksi etsiessään uutissovelluksia tai selatessaan suosituimpien sovellusten listoja.

Myös mobiilisivusto toimii kanavana, josta kiinnostuneet käyttäjät ohjataan

sovelluskauppoihin lataamaan uutispalvelun sovellus. Kuvassa 5.1 on esitetty Ilta-Sanomien iPhone-sovellus Applen AppStore-ohjelmistokaupan hakutuloksissa sekä mobiilisivustolla sovellusta mainostava banneri.

Natiivisovellusten rooli on tarjota lisätoimintoja, joita sivustoilla ei pysty tekemään. Tärkeimpiä teknisiä etuja uutispalvelun kannalta ovat *notifikaatiot* ja *pyyhkäistävät näkymät*. Tavoitteena on myös, että sovellukset tarjoavat paremman käyttökokemuksen kuin selaimen kautta käytettävä palvelu.

Notifikaatioiden avulla käyttäjille pystytään nopeasti välittämään tieto ajankohtaisista uutisista ja siten edistämään palvelun säännöllistä käyttöä. Pyyhkäistävien näkymien avulla puolestaan sovelluksiin voidaan tuoda kokosivun mainoksia, jotka ovat kaupallisesti tärkeä mainosformaatti. Kaupallisia eroja käsitellään tarkemmin kappaleessa 5.2.5.

Käyttöönottokynnys

Natiivisovellusten käyttöönottokynnys on suurempi kuin web-sivustolla. Natiivisovellus vaatii sovelluksen etsimistä ohjelmistokaupasta ja asentamista ennen palvelun käyttöönottoa. Tämä aiheuttaa merkittävän kynnyksen palvelun käyttöönottoon. Käyttäjän pitää olla kiinnostuneempi palvelusta asentamiseen sovelluksen kuin avatakseen web-sivun.

Natiivisovellusten käyttöönottokynnyksessä on havaittavissa eroja eri mobiilialustojen ja käyttäjäryhmien välillä. Käyttöönottokynnys riippuu siitä, miten käyttäjä on tottunut mobiililaitettaan käyttämään. Käyttötottumuksissa ja käyttäjäprofiileissa on puolestaan havaittavissa selkeitä eroja mobiilialustojen välillä.

On olemassa käyttäjäryhmiä, jotka ei löydä verkkosaitteja, ja on olemassa käyttäjäryhmiä jotka eivät löydä mobiilisovelluksia. On käyttäjiä jotka löytävät kummatkin, mutta me katsotaan että kumpikin tukee toisiaan ja olisi tosi vaikeaa tehdä jotain mobiili-tuotetta ilman että ne kumpikin olisi olemassa.

- Sanoma-konsernin mobiili- ja laitestrategian johtaja

Sanoma-konsernin palveluissa on havaittu, että alustoista Applen iOS on profiloitunut sovelluskeskeisimmäksi, joten iOS-käyttäjille kynnys sovelluksen asentamiseen on pienin. He ovat tottuneet että useimmille palveluille on saatavilla sovellus, joten he etsivät tarvitsemansa palvelut AppStoresta enemmän kuin selaimella. Androidin ja erityisesti Windows Phonen käyttäjät ovat puolestaan tottuneet siihen, että sovelluksia on heikommin saatavilla.

Mobiilisivustot ovat heille tutumpia ja he saattavat käyttää sivustoa, vaikka sovellus olisikin saatavilla.

Käytön säännöllisyys ja sovellusten priorisointi

Sanoma-konsernin uutissovellusten käyttötilastoista on havaittu, että natiivisovelluksia käytetään säännöllisemmin kuin web-sivustoa. Natiivisovelluksissa tehdään myös enemmän sivulatauksia käyttösessiota kohden. Tämän takia natiivisovellukset ovat käyttäjää kohden kaupallisesti tuottoisampia kuin web-sivustot. Kanavien eroja kaupallisten tavoitteiden kannalta käsitellään tarkemmin kappaleessa 5.2.5.

Koska natiivisovellusten käyttö on säännöllisempää ja kaupallisesti tuottoisampaa, niin käyttäjien halutaan ensisijaisesti käyttävän sovelluksia. Sovelluskäyttöä pyritään edistämään tarjoamalla sovelluksiin lisäarvoa tuottavia toimintoja. Uudet ominaisuudet pyritään tuomaan ensin sovelluksiin ja vasta sen jälkeen mobiilisivustolle.

Esimerkiksi Ilta-Sanomien kommentointi-toiminnallisuus julkaistiin ensin natiivisovelluksissa ja sovellusten uutta ominaisuutta mainostettiin mobiilisivustolla. Myöhemmin kommentointi-toiminnallisuus julkaistiin myös mobiilisivustolle.

5.2.2 Sisällön yhteneväisyyden ja toimivuuden ylläpito on ongelma natiivisovelluksissa

Sisällön yhteneväisyys ja toimivuus kaikissa kanavissa sekä kokonaisuuden hallittavuus määriteltiin kappaleessa 5.1.1 yhdeksi uutispalvelun tärkeimmistä vaatimuksista.

Sisällön yhteneväisyyden ja hallittavuuden saavuttaminen natiivisovelluksilla on huomattavasti työläämpää ja ongelmallisempaa kuin web-sivustolla. Natiivisovellukset joudutaan toteuttamaan erikseen jokaiselle alustalle, jolloin niiden pitäminen yhteneväisinä vaatii enemmän kehitystyötä ja aikaa.

Tilanteesta tekee erityisen ongelmallisen natiivisovellusten vanhat versiot, joita käyttäjät eivät päivitä tuoreimpaan versioon. Tällöin palvelusta on samaan aikaan käytössä useita eri versioita, jolloin eri käyttäjien näkemä sisältö ja toiminnallisuudet eivät ole yhteneväisiä. Tämä on kokonaisuuden hallittavuuden kannalta suuri ongelma. Vanhoihin versioihin liittyviä ongelmia käsitellään tarkemmin kappaleessa 5.2.3.

Mobiilisivuston sisältö on helpompi pitää yhteneväisenä muiden kanavien

kanssa, sillä sivustoa voidaan helpommin ja nopeammin päivittää vastaamaan uusia ominaisuuksia ja muutoksia. Myös sisällön toimivuuden osalta mobiilisivusto on etuasemassa natiivisovelluksiin verrattuna, sillä HTML-pohjaiset sisältökomponentit on helpompi saada toimimaan osana mobiilisivustoa kuin natiivisovelluksia.

Ilta-Sanomien sisällön yhteneväisyys mobiilikanavissa

Ilta-Sanomien mobiilipalveluissa sisällön yhteneväisyys ja toimivuus eri mobiilikanavissa ovat nykyisten kanavien suurimpia ongelmia. Varsinainen artikkelisisältö on pääosin saatu näkyviin kaikissa kanavissa, mutta erityisesti etusivun yhteneväisyydessä ja monimediaisen sisällön toimivuudessa on ongelmia. Kanavista mobiilisivusto on tällä hetkellä sisällöltään ja toimivuudeltaan yhteneväisempi pääsivuston kanssa kuin natiivisovellukset.

Mobiilikanavien sisällön ja toimivuuden erot ovat aiheuttaneet tilanteen, jossa on hankalaa hahmottaa missä kanavissa mikäkin sisältö näkyy ja miten eri toiminnallisuudet toimivat. Kuten kappaleessa 5.1.1 todettiin, niin uutistoitituksen näkökulmasta mobiili hahmotetaan yhtenä kanavana. Ongelmallisia tilanteita aiheutuu siitä, kun sisältö ei toimikaan yhteneväisesti kaikissa mobiilikanavissa.

Sisällön yhteneväisyyttä ja toiminnallisuuksien toimivuutta on jatkuvasti kehitetty eteenpäin, mutta ongelmaa pahentaa natiivisovellusten vanhat versiot. Vanhoissa versioissa sisällön rakenne saattaa olla erilainen tai jotkin toiminnallisuudet eivät toimi oikein. Pahimmassa tapauksessa jokin uusi sisältötyyppi saattaa jopa kaataa vanhan sovellusversion. Natiivisovellusten vanhoja versioita käytetään laajasti, koska monet käyttäjät eivät päivitä sovelluksiaan.

Etusivun yhteneväisyys eri kanavissa

Suurimmat ongelmat Ilta-Sanomien sisällön yhteneväisyydessä liittyvät palvelun etusivuun. Mobiilisivustolla etusivun pääkaistan sisältö vastaa melko hyvin pääsivuston sisältöä. Eri alustojen mobiilisovelluksissa sen sijaan on enemmän eroja verrattuna pääsivuston sisältöön.

Etusivun suurimmat erot liittyvät sivun rakenteeseen ja uutisten hierarkian säilyttämiseen. Etusivulla näytettävät osiot ja niiden sisältämien *nostojen* määrät vaihtelevat sovellusten välillä.

Ongelmien taustalla on se, että sisällön haussa käytetyt parametrit on

määritelty sovelluksissa, jolloin ne on asetettu erikseen kuhunkin sovellukseen. Esimerkiksi etusivulla näytettävät osiot ja niiden sisältämien artikkelien määrä on määritelty sovelluksessa. Tällöin niihin ei voida tehdä muutoksia ilman koko sovelluksen päivittäistä.

Mobiilisivuston osalta parametrit on myös määritelty ohjelmakoodissa, mutta koska sivuston päivittäminen on helpompaa, niin vanhojen versioiden ylläpitämisen ongelmaa ei ole. Lisäksi mobiilisivustolla on erityisesti panostettu etusivun sisällön saamiseen pääsivustoa vastaavaksi, joten tilanne on parempi kuin natiivisovellusten osalta.

Toimitus haluaisi myös entistä enemmän muokata etusivun rakennetta uutis-tilanteen mukaan, mutta tätä ei nykyisellä arkkitehtuurilla ole mahdollista tuoda mobiilikanaviin. Muutokset rakenteeseen vaatisi erilliset päivitykset kaikkiin kanaviin.

Monimediaisen sisällön toimivuus

Monimediaisen sisällön - kuten videoiden ja erilaisten sisältökomponenttien - toimivuus nykyisissä mobiilikanavissa on vielä puutteellista. Tämä aiheuttaa ongelmia uutistoimitukselle, koska sisällön toimivuutta ei voida taata.

Videoissa ongelmia on aiheuttanut mobiilialustojen videotuen erot ja tarve videoiden konvertoimiselle (engl. *encoding*) kunkin alustan tukemaan muotoon, sillä alustojen tukemat videoformaatit vaihtelevat. Tämä vaatii taustajärjestelmiä, jotka osaavat tuottaa videoista alustojen mukaisia versioita. Lisäksi videoiden konvertoiminen on melko raskas prosessi, joten se vie aikaa ja videot eivät välttämättä ole tarpeeksi nopeasti saatavilla kaikille eri laitteille.

Videoissa olisi tärkeää saada näkymään mainokset myös mobiilikanavassa, jotta niiden tuottaminen olisi kaupallisesti järkevää. Tässä eri kanavien tekniset erot aiheuttavat lisävaivaa. Mainosten toimivuutta käsitellään tarkemmin kappaleessa 5.2.5

Sisältökomponenttien osalta ongelmia on aiheuttanut erityisesti natiivisovellukset, joissa tuki kullekin uudelle widgetille pitää toteuttaa sovellukseen. Mobiilisivustolla widgettien tukeminen on ollut helpompaa, sillä ne perustuvat useimmiten web-teknologioihin.

5.2.3 Web-sivuston muutosten tekeminen on nopeampaa ja päivitykset hallittuja

Web-sivuston ja natiivisovellusten merkittävä ero uutispalvelun kannalta on muutosten tekemisessä ja niiden hallittavuudessa. Web-sivustolla on mahdollista tehdä nopeita muutoksia palveluun ja ne päivittyvät välittömästi kaikille käyttäjille. Natiivisovellusten osalta muutosten saattaminen käyttäjille on huomattavasti hitaampi prosessi ja kokonaisuuden hallinta hankalampaa.

Web-sivustolla muutokset voidaan viedä tuotantoon erittäin nopeasti. Kaikki sivuston kävijät käyttävät aina tuoreinta versiota palvelusta, joten versioiden hallinta on helppoa. Pieniä alustakohtaisia optimointeja sivustoon tarvitaan jonkin verran, mutta pääasiassa sivuston muutosten tekeminen vaatii vain yhden koodipohjan muokkaamista.

Natiivisovelluksissa puolestaan muutokset pitää tehdä erikseen kunkin alustan sovellukseen, hyväksyttää uusi versio ohjelmistokaupassa ja odottaa, että käyttäjät päivittävät sovelluksen uuden version laitteeseensa. Sovellusten hyväksyttäminen Applen ja Microsoftin ohjelmistokaupoissa kestää noin viikon.

Googlen ohjelmistokaupassa vastaavaa hyväksyttämistä ei aikaisemmin vaadittu, mutta myös Google otti keväällä 2015 käyttöön sovellusten hyväksymisprosessin¹. Googlen hyväksymisprosessi on kuitenkin merkittävästi Applen ja Microsoftin vastaavia prosesseja nopeampi. Googlen tavoitteena on, että hyväksyntään kuluu ainoastaan tunteja, ei päiviä tai viikkoja.

Uutispalvelun vaatimus nopeista, jopa samana päivänä tehtävistä muutoksista (ks. 5.1.2) on käytännössä mahdollista saavuttaa vain web-tekniikoilla. Natiivisovellusten kokonaisuudessa isojen muutosten sykli on viikkoja tai kuuksia johtuen usean eri sovelluksen päivittämisestä, testauksesta ja ohjelmistokauppojen prosesseista.

Käyttäjät eivät päivitä natiivisovelluksia

Natiivisovellusten isona ongelmana on, että käyttäjät päivittävät sovelluksia hitaasti - osa ei koskaan - jolloin eri käyttäjät näkevät sisällön eri tavoin ja sovelluksen vanhoja versioita pitää tukea rajapinnoissa pitkäänkin.

Natiivisovelluksissa uusien toimintojen roll-out kestää aina. Ensinnäkin täytyy tehdä se sovellus kaikille alustoille uusien toimin-

¹<http://android-developers.blogspot.fi/2015/03/creating-better-user-experiences-on.html>

tojen osalta, sitten se pitää hyväksyttää sovelluskaupassa ja kun se hyväksytään, niin silloinkin kuitenkin vain pieni osa käyttäjistä päivittää sen heti. Ja seuraavat 50% päivittää sen joskus ja 20% ei päivitä koskaan.

- Sanoma-konsernin mobiili- ja laitestrategian johtaja

Päivitysten jakelun tilanne on hieman parantunut kun sovellusten automaattiset päivitykset ovat tulleet mahdollisiksi Applen iOS-alustan 7-versiosta alkaen ja Androidin Play-kaupan uusissa versioissa. Automaattiset päivitykset eivät kuitenkaan ole oletuksena päällä kummassakaan alustassa. Windows Phonella sovellusten automaattipäivitystä ei ole.

Päivitysten lataaminen ja asentaminen käyttää paljon puhelimen resursseja, joten valmistajat ovat asettaneet automaattiselle päivitykselle rajoituksia, kuten että laitteen pitää olla latauksessa ja WLAN-verkossa. Lisäksi mahdolliset muutokset sovelluksen käyttöoikeuksissa vaativat luonnollisesti hyväksynnän käyttäjältä, jolloin päivitys pitää edelleen hyväksyä manuaalisesti.

Ilta-Sanomien tapauksessa erityisen ongelmallisia sovelluksia päivitysten kannalta ovat Nokian Suomessa myymiin Lumia-puhelimiin esiasennetut versiot Ilta-Sanomien Windows Phone -sovelluksesta. Sovellukset voisi päivittää Windowsin sovelluskaupan kautta, mutta huomattava osa käyttäjistä ei ole sovellusta koskaan päivittänyt.

Windows Phonen osalta todennäköinen syy sovellusten päivittämättömyyteen lienee käyttäjien tottumattomuus älypuhelimien ohjelmistokauppoihin. Nokian Lumia-puhelimia on ostettu Suomessa paljon ensimmäiseksi älypuhelimeksi ja monelle käyttäjälle sovelluskaupan käyttö ja sovellusten päivittäminen on vielä tuntematon konsepti.

Muutoksien laadunvalvonta ja testaaminen

Vaadittava laadunvalvonnan ja testauksen määrä eroaa myös web-sivuston ja natiivisovellusten välillä. Natiivisovellukset tulee testata erittäin hyvin ennen niiden lähettämistä ohjelmistokauppaan. Web-sivuston muutokset on toki myös testattava hyvin ennen tuotantoonvientiä, mutta mahdollisissa ongelmatilanteissa korjausten päivittäminen on paljon helpompaa ja nopeampaa kuin natiivisovelluksissa. Tällöin testauksessa havaitsematta jääneet ongelmat eivät aiheuta yhtä suurta riskiä.

Natiivisovellusten uusien versioiden testaus tulee tehdä huolella, sillä ongelmallisiin versioihin liittyy iso riski. Jos käyttäjille pääsee jakeluun versio jossa

on ongelmia, niin korjatun version toimittaminen ohjelmistokaupan prosessien läpi on hidasta - vaikka korjauksen tekeminen itsessään olisi hyvin nopea työ. Tällä välin käyttäjät joutuvat käyttämään rikkiäistä versiota sovelluksesta ja on olemassa riski, että käyttäjä lopettaa palvelun käytön kokonaan.

Korjauksen julkaisun jälkeenkään osa käyttäjistä ei päivitä sovelluksiaan ja jää käyttämään ongelmallista versiota, jolloin heidän käyttämä palvelu on puutteellinen. Lisäksi ongelmallista versiota joudutaan tukemaan taustajärjestelmissä ja mahdollisesti tekemään epäoptimaalisia ratkaisuja sen takia.

Hyvästä testauksesta huolimatta joskus käy niin, että jotkin ongelmat uudessa versiossa ilmenevät vasta ison käyttäjämäärän myötä. Tämän takia on tärkeää, että sekä natiivisovelluksissa että web-sivustolla on järjestelmät, joilla virhetilanteita voidaan seurata ja raportoida kehittäjille.

Natiivisovellusten osalta eri mobiilialustat keräävät automaattisesti sovellusten crash-raportteja, kun sovellus kaatuu. Lisäksi voidaan käyttää palveluita, jotka lähettävät sovellusten kehittäjälle tietoa muista virhetilanteista. Web-sivustolla backend-puolen virheitä voidaan havaita palvelimien logeista, mutta myös käyttäjän selaimessa tapahtuvat virheet kannattaa kerätä sopivalla front-end-monitorointi järjestelmällä.

AB-testaus

Web-sivuston nopea muokattavuus mahdollistaa myös uusien ominaisuuksien ja eri design-vaihtoehtojen testaamisen loppukäyttäjillä. Web-sivustolla voidaan helposti tehdä *AB-testausta*, jossa eri käyttäjille näytetään eri versiot jostakin palvelun ominaisuudesta ja analytiikkajärjestelmien avulla mitataan minkälaisia vaikutuksia tällä on palvelun käyttöön. Saatujen tuloksien avulla voidaan tehdä päätöksiä tuotantoon otettavista ominaisuuksista ja palvelun jatkokehittämisestä.

Ilta-Sanomien mobiilisivustolla on onnistuneesti hyödynnetty AB-testausta uusien ominaisuuksien ja sisällön rakenteen testaamisessa. Mobiilisivuston helppo päivitettävyyden on mahdollistanut, että AB-testejä on voitu tehdä hyvin nopealla syklillä. Esimerkiksi artikkelisivulla artikkelin jälkeen olevien linkkilistojen kokoonpanoa ja järjestystä on testattu aktiivisesti ja löydetty parhaat tulokset saavuttava rakenne. Mobiilisivustolle on toteutettu järjestelmä, jolla AB-testejä voidaan ajastaa ja ajaa useita eri testejä kerrallaan. Järjestelmä on yhdistelmä palvelinpään logiikkaa ja päätelaitteissa JavaScriptillä tehtävää logiikkaa.

Natiivisovellusten osalta AB-testauksen toteuttaminen on huomattavasti hankalampaa ja rajoittuneempaa kuin web-sivustolla. Testattavat versiot ominaisuuksista pitää olla etukäteen toteutettuna sovelluksissa, jotta niitä voidaan kytkeä päälle ja mitata. Pelkän sisällön osalta AB-testaus on hieman helpompaa, sillä taustajärjestelmistä voidaan tarjota eri käyttäjille erilaisista sisältöä, jolloin AB-testauksen logiikka voidaan pitää taustajärjestelmän puolella.

Esimerkiksi LinkedIn-palvelu on toteuttanut natiiveissa mobiilisovelluksissaan sisällön AB-testausta. Heidän ratkaisunsa perustuu siihen, että sovelluksiin on toteutettu mahdollisimman geneerisiä *templateja*, joita käyttäen voidaan taustajärjestelmissä räätälöidä kullekin käyttäjälle näytettävä sisältö. Kunkin templatien tarvitsemat datat tulevat taustajärjestelmästä suoraan sopivassa muodossa ja varsinaisessa sovelluksessa on mahdollisimman vähän logiikkaa sisältöön liittyen. (Gupta, 2014)

5.2.4 Käyttökokemuksessa natiivisovellukset ovat alustan mukaisia

Brändin mukainen ulkoasu ja hyvä alustan mukainen käyttökokemus todettiin kappaleessa 5.1.3 uutispalvelun tärkeäksi vaatimukseksi. Brändin mukainen ulkoasu on toteutettavissa hyvin sekä natiivi- että web-tekniikoilla. Alustan mukainen käyttökokemus sen sijaan on toteutettavissa paremmin natiivisovelluksissa.

Natiivisovelluksilla voidaan tarjota paras mahdollinen käyttökokemus kullekin mobiilialustalle. Natiivisovelluksissa alustan konventioiden noudattaminen on helppoa. Alustat tarjoavat valmiita käyttöliittymäkomponentteja sovelluksiin. Niitä hyödyntämällä saadaan toteutettua sulava ja hyvin toimiva käyttöliittymä kyseiselle alustalle.

Natiivisovellusten hyvä käyttökokemus muodostuu monista pienistä yksityiskohdista. Sisällön lataukset ja vaihdokset tapahtuvat sulavasti. Sovellus reagoi nopeasti käyttäjän interaktioihin. Pieniä *transitioita* ja animaatioita käytetään sulavan ja miellyttävän vaikutelman luomiseen.

Web-sivustolla ei ole mahdollista saavuttaa natiivisovelluksen kaltaista alustakohtaista käyttökokemusta, mutta omassa kategoriassaan sivuston käyttökokemus on mahdollista tehdä miellyttäväksi ja sujuvaksi. Web-tekniikat soveltuvat lähtökohtaisesti erittäin hyvin uutispalvelun sisällön esittämiseen.

Web-tekniikoilla ei kannata yrittää matkia natiivin omaista käyttöliittymää.

Sen sijaan tulee noudattaa yleisiä web-sivustojen käyttökonventioita, joihin käyttäjät ovat tottuneet. Sivuston ei tarvitse edes yrittää näyttää alustan mukaiselta, sillä käyttäjien odotukset selaimessa näytettävälle sisällölle eroavat natiivisovelluksista.

Ilta-Sanomien sovellukset ja sivusto tarjoavat hyvän käyttökokemuksen, mutta parannettavaakin on

Ilta-Sanomilla sekä natiivisovellusten että mobiilisivuston käyttökokemus on saatu miellyttäväksi. Natiivisovellukset noudattavat melko hyvin alustojen konventioita. Mobiilisivusto puolestaan noudattaa hyväksi todettuja mobiilisivustojen konventioita. Ilta-Sanomien brändi tulee hyvin esiin molemmissa.

Natiivisovellusten käyttöliittymien pitäminen alustojen muuttuvien käyttöliittymien mukaisina on tuottanut hieman ongelmia iOS ja Android -alustoissa. Molempien alustojen käyttöliittymissä on tapahtunut iso muutos, iOS:ssa versiossa seitsemän ja Androidissa versiossa neljä.

Natiivisovelluksissa on myös ollut ongelmia vierityspefformanssin kanssa. Erityisesti etusivun vierittäminen on välillä takkuisaa. Tämä vaikuttaa kielteisesti käyttökokemukseen. Vieritysongelmaa on ollut erityisesti iOS ja Android -sovelluksissa.

Ilta-Sanomien etusivu on raskas ja sisältää paljon erityyppisiä elementtejä, jolloin natiivit komponentit eivät ole pystyneet piirtämään sitä parhaalla mahdollisella sulavuudella. Natiivit listanäkymät soveltuvat parhaiten tasalaatuisen sisällön näyttämiseen, jolloin ne voivat kierrättää taustalla samoja elementtejä. HTML soveltuu paremmin vaihtelevan sisällön esittämiseen, joten web-sivustolla samanlaista vieritysongelmaa ei ole ollut.

5.2.5 Kaupalliset tavoitteet: natiivisovellukset tuottoisampia, uusissa mainostyypeissä ongelmia

Uutispalvelun kaupalliseksi tavoitteeksi identifioitiin kappaleessa 5.1.4 mainosinventaarin kasvattaminen laajan käyttäjäpohjan, säännöllisen käytön sekä uusien kiinnostavien mainostyyppien avulla.

Sekä natiivisovellukset että web-sivusto ovat kaupallisen tavoitteen saavuttamisen kannalta tärkeitä. Mobiilisivusto mahdollistaa laajan käyttäjäpohjan palvelemisen, mutta natiivisovellukset ovat käyttäjää kohden tuottoisampia kuin mobiilisivusto. Uusia mainostyyppiejä voidaan tarjota molemmissa kanavissa, mutta natiivisovelluksissa niiden kanssa on enemmän ongelmia kuin

sivustolla.

Laajaa käyttäjäpohjaa voidaan palvella tehokkaimmin web-sivuston avulla

Mahdollisimman laajaa käyttäjäpohjaa on tehokkainta palvella mobiilikäyttöön optimoidun web-sivuston avulla. Natiivisovellukset joudutaan tekemään erikseen jokaiselle alustalle, joten laajan käyttäjäpohjan palveleminen sovellusten kautta vaatii huomattavasti enemmän panostuksia palvelun kehittämiseen.

Kuten kappaleessa 5.2.1 todettiin, uutispalvelulla tulee olla kanava, josta jokainen potentiaalinen käyttäjä pääsee palvelua käyttämään. Web-sivusto on tehokkain tapa toteuttaa palvelu, joka on käytettävissä mahdollisimman monella mobiililaitteella.

Web-sivuston toteuttaminen niin, että se toimii luotettavasti kaikilla mobiililaitteilla vaatii jonkin verran lisätyötä alustakohtaisten erojen huomioimiseksi. Erityisesti vanhempien ja heikkotehoisempien peruspuhelinten huomioiminen vaatii monenlaisia optimointeja. Työmäärä on kuitenkin huomattavasti pienempi kuin erillisten natiivisovellusten toteuttamisessa.

Natiivisovellukset ovat käyttäjää kohden tuottoisampia

Keskimääräinen tuotto käyttäjää kohden (*ARPU*, *average revenue per user*) on parempi natiivisovelluksissa. Tähän vaikuttaa se, että sovelluksia käytetään säännöllisemmin ja enemmän kerrallaan, joten käyttäjää kohden saadaan enemmän mainosnäyttöjä. Lisäksi natiivisovelluksissa on kalliimpia mainospaikkoja kuin web-sivustolla.

Erityisen tuottoisia mainospaikkoja ovat natiivisovellusten koko sivun mainokset. Mainostajat ovat valmiita maksamaan koko sivun mainoksista enemmän kuin sisällön joukossa olevista bannerimainoksista, sillä koko sivun mainoksen huomioarvo on paljon suurempi.

Koko sivun mainoksia voidaan sijoittaa natiivisovellusten vaakasuuntaisten *karusellinäkymien* väliin sekä pyyhkäistävien artikkelinäkymien väliin. Sovelluksissa pitää edistää pyyhkäisemällä tapahtuvaa navigointia, jotta mainoksia saadaan näytettyä käyttäjille.

Uusien mainostyyppien ongelmat

Uusien mainostyyppien saaminen toimimaan on osoittautunut ongelmalliseksi erityisesti natiivisovelluksissa. Uusia mainostyyppijä ovat esimerkiksi interaktiiviset kosketusmainokset sekä video-mainokset. Esimerkiksi pyyhkäistävä 3D-kuutio on suosittu interaktiivinen mainostyyppi, josta mainostajat ovat kiinnostuneita. Mainostyyppijä käsiteltiin tarkemmin kappaleessa 5.1.4.

Interaktiivisten mainosten toimivuudessa suurimmat ongelmat ovat Windows Phone -alustalla. Interaktiiviset mainokset ovat HTML-pohjaisia ja niissä käytetään usein moderneja *CSS-transitioita* sekä kosketuseleitä, joiden tuki Windows Phonen selainkomponentissa on huonompi kuin iOS ja Android -alustoilla. Mainosten toimimattomuus Windows Phone -laitteissa on iso kaupallinen ongelma erityisesti suomalaisille medioille, sillä Windows Phone on Suomessa suosittu alusta.

Myös mainosten hallinnan kannalta on ongelmallista, että mainokset eivät toimi luotettavasti kaikissa sovelluksissa. Tällöin mainokset joudutaan erikseen kohdistamaan niihin alustoihin ja versioihin, joissa ne toimivat.

5.2.6 Natiivisovellukset ovat nopeampia ja luotettavampia

Kappaleessa 5.1.5 esitetty nopeuden ja luotettavuuden vaatimus edellyttää, että uutispalveluun julkaistu sisältö on mahdollisimman nopeasti käyttäjien saatavilla, palveluiden käyttö on nopeaa ja palvelut toimivat kaikissa tilanteissa luotettavasti.

Natiivisovelluksilla on monilta osin paremmat mahdollisuudet vastata nopeuden ja luotettavuuden vaatimuksiin kuin web-sivustoilla. Natiivisovelluksissa pystytään esimerkiksi välittämään tieto uusista uutisista nopeasti käyttäjille notifikaatioiden avulla. Välimuistin käytön ja pienempien tiedonsiirtomäärien avulla natiivisovelluksista on mahdollista tehdä luotettavampia ja nopeampia käyttää myös huonoissa verkkoyhteyksissä.

Toisaalta natiivisovellusten pitämisessä luotettavina ja laadukkaina on isompi työ erillisten koodipohjien takia ja sovellusten vanhat versiot voivat myös aiheuttaa ongelmia. Seuraavaksi käydään tarkemmin läpi natiivisovellusten ja web-sivustojen eroja nopeuden ja luotettavuuden kannalta.

Notifikaatioilla saadaan lähetettyä tieto uutisista nopeasti käyttäjille

Uutisten välittämisessä mahdollisimman nopeasti käyttäjien tietoon ovat olennaisessa osassa notifikaatiot, joilla käyttäjän laitteeseen voidaan lähettää *push*-viesti välittömästi kun käyttäjän tilaaman kategorian uutinen on julkaistu.

Notifikaatioita voi lähettää natiivisovelluksiin mobiilialustojen tarjoamien notifikaatiopalveluiden ja -rajapintojen kautta. Jokaisella alustalla on oma järjestelmänsä notifikaatioiden lähettämiseen. Viestien välittäminen uutispalvelun taustajärjestelmistä eri alustojen notifikaatiopalveluihin pitää toteuttaa erikseen kullekin alustalle tai käyttää valmista integraattoripalvelua.

Push-notifikaatioiden lähetysmahdollisuus on tulossa myös web-selaimiin, mutta ei ole tämän työn kirjoitushetkellä vielä laajasti saatavilla. Avoimen *Push API* -spesifikaation² standardointiprosessi on vasta aikaisessa vaiheessa, mutta Google³ ja Mozilla⁴ ovat jo implementoimassa spesifikaation toteutusta Chrome ja Firefox-selaimiin. Mobiililaitteissa osittainen toteutus on saatavilla uusimmassa Chrome-selaimessa Android-alustalla. Apple puolestaan on julkaissut oman suljetun *Safari Push Notifications*⁵ -toteutuksen, mutta se ei ole saatavissa mobiililaitteisiin.

Tällä hetkellä notifikaatiot ovat natiivisovellusten suuri etu ja ne nähdään yhtenä tärkeimmistä syistä julkaista uutispalvelun natiivisovelluksia. Notifikaatiot edistävät uutispalvelun säännöllistä käyttöä ja vaikuttavat osaltaan natiivisovellusten web-sivustoa suurempaan käyttötiheyteen, jota käsiteltiin kappaleessa 5.2.1. Tulevaisuudessa kun notifikaatioiden tuki myös web-selaimissa yleistyy, niin selaimissa toimivien palveluiden ja natiivisovellusten erot kaventuvat tältä osin.

Välimuisti sekä tiedonsiirtomäärän optimointi nopeuttavat käyttöä ja parantavat toimintaa myös huonoilla verkkoyhteyksillä

Natiivisovelluksissa on web-sivustoja kattavammat mahdollisuudet nopeuttaa palvelun käyttöä tallentamalla tietoja välimuistiin. Artikkeleita ja muita palvelun sisältöjä voidaan helposti esiladata (engl. *preload*) taustalla valmiiksi välimuistiin. Tämä nopeuttaa merkittävästi palvelun käyttöä, kun artik-

²<http://www.w3.org/TR/push-api/>

³<https://developers.google.com/web/updates/2015/03/push-notifications-on-the-open-web>

⁴https://developer.mozilla.org/en-US/docs/Web/API/Push_API

⁵<https://developer.apple.com/notifications/safari-push-notifications/>

keli tai muu sisältö voidaan tarjota nopeasti laitteen muistista sen sijaan, että se haettaisiin verkosta vasta käyttäjän avatessa artikkelia.

Välimuisti mahdollistaa myös sovelluksen käyttämisen ilman verkkoyhteyttä - eli *offline-tilassa* - niiden artikkelien ja näkymien osalta jotka on ladattu laitteen muistiin. Toisaalta välimuistiin ei ole järkevää hakea liikaa sisältöä, ettei aiheuteta ylimääräistä tiedonsiirtoa. Esiladattavan sisällön rajaamiseen on sovelluksissa mahdollista käyttää esimerkiksi tietoa käyttäjän lukutottumuksista.

Web-sivustoilla mahdollisuudet sisällön esilataamiseen ja yhteydettömän tilan tukemiseen ovat natiivisovelluksia rajatummalla ja vaativat enemmän lisätyötä. Kattavan esilatauksen toteuttaminen vaatii käytännössä web-sivuston toteuttamista ns. *yhden sivun sovelluksena* (engl. *Single Page Application*), jossa sisältö haetaan *Ajax*-kutsuilla ja tallennetaan esimerkiksi selaimen *Local Storage*-rajapintaan⁶. (Nieminen, 2015) Uutispalveluiden web-sivustot ovat yleensä toteutettu normaaleina HTML-sivuina, jolloin tällaisen esilatauksen toteuttaminen ei ole mahdollista.

Tiedonsiirtomäärän minimointi vaikuttaa myös merkittävästi mobiilipalvelun toimintanopeuteen. Natiivisovelluksissa tiedonsiirtomäärät ovat web-sivustoja pienemmät, sillä verkosta tarvitsee hakea pelkästään varsinainen uutisdata. Web-sivuston tapauksessa verkosta haetaan jokaisen sivun HTML-rakenne ja sivuun liittyvät resurssit. Web-sivun latautumisenopeuteen pystyy vaikuttamaan erilaisilla performanssi-optimoinneilla, kuten vähentämällä erillisten kyselyiden määrää niitä yhdistelemällä.

Välimuistin käyttö ja tiedonsiirtomäärien minimointi nopeuttavat palvelun toimintaa erityisesti huonolaatuisilla verkkoyhteyksillä, joita mobiiliverkoissa usein esiintyy. Natiivisovellukset toimivatkin yleisesti web-sivustoja luotettavammin huonolaatuisilla verkkoyhteyksillä kattavamman välimuistin ja pienempien tiedonsiirtomäärien ansiosta.

Luotettavuuden ja laadukkuuden varmistaminen on työlästä

Luotettavuuden ja toimintavarmuuden osalta tutkimukseen haastatellut näkivät natiivisovellukset luotettavampana kanavana kun web-sivut. Toisaalta usean eri alustan sovellusten erillisten koodipohjien ylläpitäminen luotettavana ja laadukkaana vaatii paljon työtä. Sovelluksiin nähtiin kertyvän helposti teknistä velkaa, kun uusia ominaisuuksia lisätään ja koodipohjan *refaktorointiin* ei jää tarpeeksi aikaa.

⁶<http://www.html5rocks.com/en/tutorials/offline/storage/>

Natiivisovelluksissa ongelmia palvelun luotettavuudelle aiheuttavat myös sovellusten vanhat versiot. Kuten kappaleessa 5.2.3 käsiteltiin, sovellusten vanhoja versioita on käytössä käyttäjillä jotka eivät päivitä sovelluksiaan. Uudemmissa versioissa korjatut ongelmat jäävät edelleen vanhoihin versioihin ja aiheuttavat haasteita esimerkiksi taustajärjestelmien kehitykseen. Toimivan ja luotettavan palvelun takaaminen myös vanhojen versioiden käyttäjille vaatii erityistä huomiota taustajärjestelmien kehityksessä ja hidastaa uusien ominaisuuksien kehittämistä.

Web-sivuston osalta ongelmia luotettavan palvelun tarjoamiselle aiheuttavat puolestaan käyttäjien vanhat selainversiot. Vanhojen selaimien tukemisesta aiheutuu paljon lisätyötä erityisesti testaamisen muodossa ja rajoitteita uusien ominaisuuksien kehittämiseen.

5.2.7 Yhteenveto

Natiivisovellukset ja mobiili web-sivusto ovat uutispalvelun jakelukanavina erilaisia, mutta toisiaan täydentäviä ja tukevia. Mobiilisivusto on helposti saavutettavissa ilman asentamista ja mahdollistaa palveluun linkittämisen esimerkiksi sosiaalisesta mediasta. Natiivisovellukset puolestaan tarjoavat käyttäjille alustakohtaisen käyttökokemuksen ja monipuolisempia ominaisuuksia.

Mobiilisivusto nähdään välttämättömyytenä, joka uutispalvelulla tulee joka tapauksessa olla. Natiivisovellukset puolestaan edistävät säännöllistä käyttöä ja ovat kaupallisesti tuottoisampia käyttäjää kohden, joten niihin investoiminen on järkevää.

Taulukossa 5.2 on esitetty yhteenveto siitä, miten natiivisovellukset ja mobiilisivusto vastaavat kappaleessa 5.1 identifioituihin uutispalvelun vaatimuksiin. Taulukosta nähdään, että molemmilla teknologioilla on etunsa ja ongelmansa uutispalvelun toteuttamisessa.

Vaatus	Natiivisovellukset	Web-sivusto
Sisällön yhteneväisyys, toimivuus ja hallittavuus	Hankalampia pitää yhteneväisenä, koska joudutaan toteuttamaan erikseen jokaiselle alustalle.	Yhtenevä sisältö kaikissa laitteissa. HTML-pohjainen sisältö toimii hyvin.
Muokattavuus	Muutokset vaativat uuden sovellusversion lataamista. Vanhat versiot aiheuttavat ongelmia, kun kaikki käyttäjät eivät päivitä sovelluksiaan.	Muutosten tekeminen nopeaa ja hallittua. Päivittyvät välittömästi kaikille käyttäjille. Mahdollistaa helpon AB-testauksen.
Käyttökokemus ja ulkoasu	Paras mahdollinen alustan mukainen käyttökokemus.	Ei mahdollista saavuttaa alustan konventioiden mukaista käyttökokemusta.
Kaupalliset tavoitteet	Käyttäjää kohden tuottoisampia: käytetään säännöllisemmin, kalliimpia mainospaikkoja.	Mahdollistaa laajan käyttäjäpohjan palvelemisen tehokkaasti.
Nopeus ja luotettavuus	Push-notifikaatiot tuettuja. Sisällön esilataus ja offline-tuki helppo toteuttaa. Luotettavampi heikkolaatuisissa verkkoyhteyksissä.	Sisällön esilataus ja offline-tuki hankala toteuttaa. Huonompi tuki heikkolaatuisille verkkoyhteyksille.

Taulukko 5.2: Yhteenvedo natiivisovellusten ja mobiilin web-sivuston vastaamisesta uutispalvelun vaatimuksiin

5.3 Miten hybridi-sovellukset vastaavat uutispalvelun vaatimuksiin?

Tässä kappaleessa käsitellään tutkimuksen kolmatta tutkimuskysymystä: *Voiko hybridi-sovelluksilla yhdistää web-sivuston ja natiivisovellusten hyvät puolet ja vastata entistä paremmin mobiilin uutispalvelun vaatimuksiin?*

Tutkimuskysymystä tarkastellaan Ilta-Sanomien uutissovelluksesta tehtyjen hybridi-prototyyppien ja niistä saatujen kokemusten perusteella. Hybridi-prototyypit on esitelty kappaleessa 4.3.

Hybridi-tekniikoita peilataan kappaleessa 5.1 esiteltyihin uutispalvelun vaatimuksiin sekä kappaleessa 5.2 havaittuihin natiivi- ja web-tekniikoiden etuihin ja ongelmiin.

5.3.1 Sisällön yhteneväisyys ja hallittavuus merkittävien parannus natiivisovelluksiin

Sisällön yhteneväisyys ja hallittavuus on hybridi-sovellusten merkittävä etu natiivisovelluksiin verrattuna. Tutkitussa hybridi-lähestymistavassa kaikkien alustojen sovellukset käyttävät samoja web-palvelimelta tarjottavia hybridi-näkymiä. Tällöin web-palvelimelta tarjottavissa näkymissä oleva sisältö on yhteneväistä eri alustoissa ja toimii samalla tavalla alustasta riippumatta.

Kuten kappaleessa 5.2.2 käsiteltiin, yhteneväisyys ja hallittavuus on merkittävä ongelma nykyisissä Ilta-Sanomien natiivisovelluksissa. Hybridi-lähestymistavassa pystytään natiivisovelluksia paremmin ja helpommin takaamaan, että toimituksen tekemät sisällöt näkyvät kaikissa sovelluksissa siten kuin toimittaja on ajatellut. Tämän ansiosta sisällön hallittavuus on merkittävästi parempaa kuin natiivisovelluksissa.

Hybridi-lähestymistavan todettiin haastatteluissa sopivan hyvin myös varsinaisen artikkelisisällön näyttämiseen, vaikka tätä ei hybridiprototypeissa toteutettukaan. Toimitusjärjestelmässä tuotettu artikkelin leipäteksti on muodoltaan HTML:ää, joten sen näyttäminen web-näkymässä on helpompaa kuin prosessointi natiivi-elementeiksi. Osassa nykyisiä natiivisovelluksia käytetäänkin web-näkymiä artikkelin näyttämiseen. Tämä on toiminut hyvin ja aiheuttanut vähemmän ongelmia kuin HTML:n konvertointi natiivielementeiksi.

5.3.2 Nopeat muutokset mahdollisia hybridinäkymiin, mutta kokonaisuuden hallintaan on kiinnitettävä huomiota

Hybridi-lähestymistavassa web-sivustojen etu nopeiden ja hallittujen muutosten tekemisessä (ks. kappale 5.2.3) on mahdollista tuoda myös sovelluksiin. Muutosten tekeminen on hybridi-sovelluksissa huomattavasti ketterämpää kuin natiivisovelluksissa.

Hybridi-sovelluksissa voidaan tehdä muutoksia web-palvelimelta tarjottaviin näkymiin. Muutokset web-näkymiin tehdään palvelinpäässä eikä uutta versiota natiivisovelluksesta tarvita. Muutosten hallinta on tällöin helppoa, sillä päivitetty näkymä on heti käytössä kaikilla käyttäjillä.

Isommat muutokset sovelluksen rakenteeseen, toimintalogiikkaan tai ulkoasuun vaativat myös hybridi-lähestymistavassa muutoksia sovelluksen natiivikoodiin ja uuden natiivisovellusversion tekemistä. Hybridi-sovellukset eivät välttämättä vähennä olennaisesti julkaistavien sovellusversioiden määrää. Sovellusversio- ja alustakohtaisesta hajonnasta ei päästä kokonaan eroon.

Riskinä nähtiin, että sovelluskokonaisuus muuttuu entistä monimutkaisemmaksi, kun julkaistuna on monta sovellusversiota ja monta hybridinäkymiä tarjoavan taustajärjestelmän versiota. Yhtä sovellusversiota kohden saattaa tarvita sekä natiivi- että taustajärjestelmän version.

Riskin välttämiseksi kokonaisuuden hallintaan pitää kiinnittää erityistä huomiota. Yhteistyön natiivisovellusten sekä taustajärjestelmien kehittäjien välillä tulee olla saumatonta ja uusien ominaisuuksien suunnittelu pitää tehdä huolellisesti huomioiden sekä natiivitoteutus että taustajärjestelmät.

5.3.3 Käyttökokemuksen takaaminen ja alustakohtaisen ulkoasun toteuttaminen haasteellista

Haastatteluissa todettiin, että natiivisovelluksen käyttöliittymän pitää olla kunkin mobiilialustan käyttöliittymäkonventioiden mukainen, jotta sovelluksen käyttökokemus säilyy hyvänä. Tästä syystä sovelluksen ylätasen käyttöliittymä kannattaa hybridi-sovelluksissakin tehdä natiivisti ja hybridi-lähestymistapaa käyttää vain rajatusti. Esimerkiksi sovelluksen valikot ja siirtymät eri näkymien välillä tulisi toteuttaa natiivisti.

Käyttökokemuksen takaamiseksi web-tekniikoilla ei kannata jäljitellä natiiveja käyttöliittymäkomponentteja ja yrittää tehdä koko sovelluksen

käyttöliittymää. Haastatteluissa nousi esiin laajasti huonoja kokemuksia hybridisovelluksista, joissa koko käyttöliittymä on tehty web-tekniikoilla.

Hybridi-lähestymistavan nähtiin sopivan parhaiten vaihtuvaa uutissisältöä näyttävien näkymien toteuttamiseen. Esimerkiksi etusivu, osiosivut ja artikkelisivut ovat mahdollisia toteuttaa hybridi-lähestymistavalla, sillä niiden sisällön on tarkoitus näkyä mahdollisimman samanlaisena kaikkien alustojen sovelluksissa. Sisällön osalta nähtiin, että palvelun ulkoasu ja käyttökokemus voi olla enemmän brändin kuin alustan mukainen.

5.3.4 Vaikutukset kaupallisiin tavoitteisiin epäselviä

Hybridi-lähestymistavan vaikutuksia uutispalvelun kaupallisiin tavoitteisiin ei saatu laajasti selvitettyä hybridi-prototyypin avulla. Kaupalliselta kannalta tärkeintä olisi huolehtia, että mainokset toimivat hybridisovelluksissa vähintään yhtä hyvin kuin natiivisovelluksissa ja että sovellusten käyttäjämäärät pysyvät ennallaan.

Hybridi-sovellusten suurimmaksi potentiaaliseksi eduksi nähtiin mahdollisuus edistää web-tekniikoihin perustuvien uusien mainosmuotojen toimivuutta eri alustoilla. Kuten kappaleessa 5.2.5 käsiteltiin, nykyisissä natiivisovelluksissa on ollut ongelmia uusien mainostyyppien toimivuuden kanssa.

Natiivisovelluksissa mainosten toimivuus eri alustojen välillä on liian eriytyntä ja monimutkaista. Hybridisovellukset voisivat yhtenäistää mainosten toimivuutta ja mainospaikkoja eri alustoilla. Myös uusien mainosmuotojen ja -paikkojen lanseeraaminen voisi olla hybridisovelluksissa helpompaa.

Kaupalliselta kannalta on myös tärkeää, että hybridi-lähestymistapa ei saisi vaikuttaa negatiivisesti kappaleessa 5.2.5 käsiteltyyn natiivisovellusten etuun säännöllisen käytön edistäjänä. Hybridi-sovelluksissa pitää huolehtia, että käyttökokemus säilyy yhtä miellyttävänä kuin täysin natiivipohjaisissa sovelluksissa ja käyttäjät käyttävät sovelluksia yhtä usein kuin aiemminkin.

5.3.5 Nopeus ja luotettavuus

Sovelluksen nopeus ja suorituskyky nähtiin haastatteluissa yhdeksi suurimmista riskeistä, joita hybridi-lähestymistapa voi aiheuttaa. Hybridi-prototyypeissä ei kuitenkaan havaittu merkittäviä ongelmia nopeuden ja suorituskyvyn kanssa. Hybridi-sovellukset ovat käyttäjän näkökulmasta natiivisovelluksia, joten käyttäjän havaitseman suorituskyvyn tulisi olla yhtä hyvä kuin vastaavissa natiivisovelluksissa.

Nopeuden osalta merkittävin hybridiprototyypeistä tehty havainto oli, että sovelluksen ensimmäisen näkymän piirtäminen voi olla hitaampaa kuin natiivisovelluksissa. Hitauden koettiin johtuvan siitä, että web-näkymän lataaminen ja piirtäminen on sovelluksen kannalta atomisempi operaatio, jota ei voida kontrolloida ja pilkkoa osiin yhtä hyvin kuin natiivissa. Natiivissa toteutuksessa voidaan helpommin piirtää elementtejä sitä mukaa kun dataa saadaan ladattua verkosta. Tämä vaikuttaa käyttäjälle nopeammalta, vaikka koko näkymän lataaminen ja piirtäminen olisikin yhtä nopeaa molemmissa toteutustavoissa.

Kattavan välimuistin ja sisällön esilatauksen toteuttaminen on hybridilähestymistavassa mahdollista, mutta työläämpää kuin natiivisovelluksissa. Esilatausten ja välimuistin hallinta tulee toteuttaa sovelluksen natiivikoodiin. Natiivikoodi mahdollistaa hybridi-sisällön taustalatauksen ja omien välimuistikäytäntöjen kehittämisen hybridi-näkymissä näytettävälle sisällölle. Myös sovelluksen sisällä käytetty selainkomponentti pitää yllä normaalia selaimen välimuistia ja huolehtii selaimesta käsin tehtyjen HTTP-kyselyiden tallentamisesta välimuistiinsa.

Selainkomponentti vaikuttaa huomattavasti hybridi-sovelluksen suorituskykyyn

Hybridi-lähestymistavan suorituskykyä natiivisovelluksiin verrattuna koettiin yleisesti rajoittavan se, että hybridi-näkymien piirtoon käytettävä selainkomponentti on ylimääräinen kerros järjestelmän ja palvelun välissä. Tämän takia selaimessa ajettava koodi on aina suorituskyvyssä natiivia jäljessä. Mobiililaitteiden web-selaimet ovat kehittyneet vauhdilla viime vuosina, mutta vastaavasti myös natiiviteknologiat kehittyvät eteenpäin.

Normaaliin web-palveluun verrattuna hybridisovellusten tekemistä helpottaa se, että mobiilialustojen natiivisovellusten selainkomponenttien erilaisten selainmoottorien ja -versioiden määrä on rajatumpi kuin yleensä web-sivustojen käyttäjäkunnassa. Tällöin selainkohtaiset erot ovat paremmin kontrollissa ja aiheuttavat vähemmän lisätyötä.

Toisaalta natiivialustojen tarjoamissa selainkomponenteissa voi olla rajoitteita, joita alustojen normaaleissa web-selaimissa ei ole. Esimerkiksi Applen iOS-alustalla järjestelmän *UIWebView*-komponentti⁷ on huomattavasti tehottomampi ja rajoittuneempi kuin alustan Safari-selain. (Podjarny, 2013) Tuoreimmissa iOS-alustan versioissa Apple on onneksi julkaissut uuden

⁷https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIWebView_Class/

WKWebView-komponentin⁸, joka perustuu uudempaan selainmoottoriin ja on huomattavasti vanhaa komponenttia tehokkaampi. (VanToll, 2014)

Myös Android-alustalla selainkomponenteissa on huomattavia eroja alustan versioiden välillä. Android-alustan versiosta 4.4 eteenpäin on käytössä tehokas Googlen *Chromium*-projektiin pohjautuva selainkomponentti⁹, mutta vanhemmissa versioissa on mukana tehottomampi Java-pohjainen selainkomponentti.

Android-sovelluksissa on mahdollista käyttää myös muuta kuin alustan mukana tulevaa selainkomponenttia. Esimerkiksi *Crosswalk*-projekti¹⁰ tarjoaa sovellukseen mukaan paketoitavaa selainkomponenttia, joka pohjautuu tuoreeseen Chromium-versioon. Tämä mahdollistaa tehokkaan selainkomponentin käyttämisen myös vanhemmissa laitteissa. Selainkomponentin paketoiminen mukaan sovellukseen myös vähentää laitekohtaisia eroja ja voi helpottaa palvelun toteutusta sekä testausta.

5.3.6 Yhteenveto

Hybridi-sovelluksilla avulla on rajoitetusti mahdollista tuoda web-teknologioiden etuja natiivisovelluksiin. Yhdistämällä web-teknologioiden etuja natiivisovellusten etuihin, voidaan vastata paremmin joihinkin uutispalvelun vaatimuksiin.

Hybridi-lähestymistavassa voidaan paremmin vastata uutispalvelun vaatimuksiin nopeudesta ja sisällön yhteneväisyydestä. Web-sivuna tehtyä näkymää voi päivittää nopeasti palvelinpäässä ja kaikki käyttäjät näkevät aina tuoreimman version näkymästä. Sisältö ja ulkoasu saadaan myös helpommin toteutettua yhteneväiseksi eri alustojen välillä.

Natiivin kaltaisia käyttöliittymäkomponentteja ei kannata lähteä matkimaan web-tekniikoilla, joten koko sovellusta ei yleensä ole syytä lähteä tekemään hybridinä, ainoastaan vaihtuvaa sisältöä näyttävät näkymät. Vaikutukset kaupallisiin tavoitteisiin ovat vielä epäselviä. Nopeuden ja luotettavuuden osalta hybridi-sovellukset pääsevät melko lähelle natiivisovelluksia.

Yhteenveto hybridi-sovellusten vastaamisesta uutispalvelun vaatimuksiin on esitetty taulukossa 5.3.

⁸https://developer.apple.com/library/ios/documentation/WebKit/Reference/WKWebView_Ref/

⁹<https://developer.chrome.com/multidevice/webview/overview>

¹⁰<https://crosswalk-project.org/>

Uutispalvelun vaatimus	Hybridi-sovellukset
Sisällön yhteneväisyys, toimivuus ja hallittavuus	Eri alustojen sovellusten sisältö saadaan helpommin yhteneväiseksi ja hallittavaksi.
Muokattavuus	Mahdollistaa web-sivuston kaltaiset nopeat muutokset sisältöön, mutta suuremmat muutokset on tehtävä natiiveina päivityksinä.
Käyttökokemus ja ulkoasu	Käyttökokemuksen takaaminen ja alustakohtaisen ulkoasun toteuttaminen haasteellista. Ylätasoinen käyttöliittymä tulee tehdä natiivisti.
Kaupalliset tavoitteet	Vaikutukset kaupallisiin tavoitteisiin epäselviä. Web-tekniikoihin pohjautuvat mainokset toimivat paremmin.
Nopeus ja luotettavuus	Push-notifikaatiot tuettuja. Sisällön esilataus ja offline-tuki mahdollista toteuttaa. Suorituskyky lähellä natiivia.

Taulukko 5.3: Yhteenveto hybridi-teknologioiden vastaamisesta tutkimuksessa identifioituihin uutispalvelun vaatimuksiin

Luku 6

Mobiilitekniologiastrategia

Tässä luvussa muodostetaan tutkimuksen tulosten pohjalta esitys mobiilin uutispalvelun teknologiastrategiaksi. Aluksi käsitellään teknologiastrategian yleisiä periaatteita. Seuraavaksi esitellään kaksi teknologista vaihtoehtoa naatiivisovellusten teknologiaavalinnan osalta.

6.1 Strategian yleiset periaatteet

6.1.1 Sisällön rakenteen ja käyttöliittymälogiikan määrittely taustajärjestelmissä

Tärkein tutkimuksessa tunnistettu strateginen periaate on, että uutispalvelun sisällön rakenteen ja käyttöliittymälogiikan määrittelyt kannattaa tehdä mahdollisimman suurelta osin uutispalvelun taustajärjestelmissä, eikä eri jakelukanavissa. Tällöin eri jakelukanavat saadaan pidettyä yhteneväisempinä ja muutosten tekeminen rakenteeseen sekä käyttöliittymälogiikkaan on helpompaa ja nopeampaa.

Taustajärjestelmissä tehtävät rakenteen ja käyttöliittymälogiikan määrittelyt mahdollistavat, että suurempi osa logiikasta saadaan jaettua eri kanavien välillä. Tällöin jakelukanavista saadaan yksinkertaisempia ja yhteneväisempiä. Lisäksi vähennetään kehityskustannuksia ja -aikaa, kun muutokset täytyy tehdä vain yhteen paikkaan.

Käytännössä rakenteen ja käyttöliittymälogiikan määrittely taustajärjestelmissä tarkoittaa, että data välitetään jakelukanaviin mahdollisimman pitkälle prosessoidussa muodossa. Tätä periaatetta noudattamalla voidaan vähentää tut-

kimuksessa havaittuja ongelmia natiivisovellusten yhteneväisyydessä ja muokattavuudessa.

6.1.2 Sekä Web-sivusto, että alustojen natiivisovellukset tarvitaan

Uutispalvelu kannattaa tarjota mobiililaitteisiin sekä mobiilioptimoituna web-sivustona, että natiivisovelluksina yleisimmille mobiilialustoille. Molemmilla jakelukanavilla on oma roolinsa ja ne tukevat toisiaan. Web-sivusto palvelee tehokkaasti laajaa käyttäjäpohjaa, samalla kun natiivisovellukset tarjoavat aktiivisille käyttäjille parhaan mahdollisen käyttökokemuksen ja lisäominaisuuksia.

Mobiililaitteille optimoidun web-sivuston rooli on tarjota kaikille käyttäjille mahdollisuus päästä nopeasti käsiksi uutispalvelun sisältöön päätelaitteesta riippumatta. Web-sivuston isoin etu on linkitettävyyden, joka mahdollistaa sisällön helpon jakamisen. Muun muassa sosiaaliset verkostot ovat tärkeä liikenteen lähde web-sivustolle.

Web-sivuston toteuttamisessa tulee huomioida sen toimivuus mahdollisimman laajalla laitejoukolla sekä laitekohtaiset erot. Kehittyneempien laitteiden mahdollisuuksia kannattaa hyödyntää ja samalla tarjota karsitummat toiminnallisuudet heikommille laitteille. Responsiivinen lähestymistapa sivuston ulkoasuun on suositeltava, jotta eri näyttökokoja saadaan palveltua mahdollisimman hyvin. Ulkoasun osalta on huomioitava laitekohtaiset rajoitukset ja hyväksyttävä se, että palvelun ulkoasua ei välttämättä ole teknisesti mahdollista toteuttaa identtiseksi jokaisella laitteella.

Natiivisovellusten rooli puolestaan on tarjota paras mahdollinen alustan mukainen käyttökokemus ja aktiivisille käyttäjille paremmat mahdollisuudet seurata uutisia esimerkiksi notifiointien avulla. Natiivisovellukset kannattaa toteuttaa yleisimmille mobiililaitteiden käyttöjärjestelmille ja huomioida kunkin alustan käyttökonventiot. Natiivisovellusten teknologiavalintoja käsitellään tarkemmin alakappaleessa 6.2.

6.1.3 Jaettu ohjelmalogiikka palveluiden ja alustojen välillä

Ohjelmalogiikkaa kannattaa mahdollisimman paljon jakaa eri palveluiden ja alustojen välillä, jotta vältetään päällekkäistä työtä ja saadaan kehitettyä palveluita tehokkaasti. Ohjelmalogiikkaa voi jakaa esimerkiksi muodos-

tamalla yhteisiä moduuleita tai kirjastoja, joita eri palvelut ja alustat voivat hyödyntää. Jaetun ohjelmalogiikan hyödyntäminen on mahdollista sekä web-palveluiden että natiivisovellusten kesken.

Uutiskonsernin eri brändien välillä kannattaa tehdä mahdollisimman laajaa yhteistyötä sovellusten kehittämisessä. Sekä parhaita käytäntöjä että varsinaista ohjelmakoodia kannattaa jakaa eri sovellusten välillä. Yleisimmistä toiminnallisuuksista voidaan rakentaa jaettuja koodikirjastoja, joita voi hyödyntää useassa eri sovelluksessa.

Eri alustoille kehitettyjen natiivisovellusten välillä kannattaa tutkia mahdollisuutta siirtää osa ohjelmalogiikasta alustariippumattomaan kirjastoon, samaan tapaan kuin esimerkiksi Dropbox on tehnyt¹. Esimerkiksi C++ ohjelmointikielellä kirjoitettu kirjasto voidaan kääntää usean eri mobiilialustan tukemaan muotoon.

6.1.4 Elinkaariajattelu: natiivisovellusten sulkeminen, päivitysilmoitukset ja rajapintojen versiointi

Uutispalvelun eri jakelukanavien kehittämistä suositellaan lähestyttävän elinkaariajattelun kautta. Erityisesti natiivisovellusten versioilla on tietty elinkaari ja elinkaaren tullessa päätökseen tulee sovellus hallitusti sulkea. Käyttäjät tulee ohjata asentamaan uusi versio sovelluksesta tai siirtymään mobiilisivuston käyttäjäksi.

Sovelluksen elinkaari tulee huomioida jo kehitysvaiheessa. Sovellukseen kannattaa rakentaa järjestelmä jolla käyttäjää voidaan informoida sovellukseen saatavilla olevista päivityksistä. Lopulta sovellusversion käyttö tulisi voida hallitusti kokonaan estää, kun sen tukeminen taustajärjestelmissä on lopetettu.

Elinkaariajattelun avulla vältetään riskiä, että vanhempien sovelluksien tukeminen taustajärjestelmissä häiritsee uusien toiminnallisuuksien kehittämistä ja uudempien ohjelmaversioiden täysipainotteista hyödyntämistä. Ennen tai myöhemmin taustajärjestelmät joka tapauksessa kehittyvät siten, että vanhat sovellusversiot eivät toimi enää kunnolla. Tällöin käyttäjän näkemä sisältö ei enää vastaa uutispalvelun toimittajien haluamaa tilannetta, jolloin on parempi ohjata käyttäjä eteenpäin.

Vanhojen versioiden tukemista voidaan toisaalta helpottaa ottamalla käyttöön versioidut rajapinnat taustajärjestelmissä. Tällöin taustajärjestelmissä voi-

¹Building for Cross Platform Apps at Dropbox: esitys Mobile @ Scale 2013 -seminaarissa. Taltiointi: <https://www.youtube.com/watch?v=S5rXCvu9-NM>

daan huomioida eri sovellusversioiden erot ja mukauttaa rajapintaa sen mukaisesti. Uusien ominaisuuksien kehittäminen helpottuu, kun niiden tarvitsema data voidaan kohdistaa vain uusille sovellusversioille, eikä tarvitse huomioida mahdollisia sivuvaikutuksia vanhoihin versioihin. Myös rajapinnan muutosten testaaminen helpottuu, kun se voidaan rajata tiettyihin sovellusversioihin.

6.1.5 Nopeuden ja laadun korostaminen

Viimeinen yleinen periaate liittyy mobiilipalveluiden nopeuden ja luotettavuuden maksimointiin kaikessa kehitystyössä. Sovellukset ja taustajärjestelmät tulee kehittää siten, että ne toimivat mahdollisimman nopeasti ja luotettavasti. Palveluissa tulee huomioida mobiililaitteiden rajoitukset esimerkiksi huonojen verkkoyhteyksien osalta ja huolehtia palvelun toimivuudesta kaikissa tilanteissa.

Natiivisovelluksissa tulee kiinnittää huomiota sovelluksen käynnistymisnopeuteen ja sisällön saamiseen näkyviin mahdollisimman nopeasti.

Mobiilikäyttöön suunnatulla web-sivustolla tulee huomioida sivujen latautumisenopeus ja tiedonsiirtomäärät. Sivun yläosan piirtonopeuteen kannattaa kiinnittää huomiota priorisoimalla ensimmäisessä näkymässä vaadittavien tietojen lataaminen. Alempana sivulla olevia tietoja kannattaa alkaa lataamaan ja prosessoimaan vasta kun ensimmäinen näytöllinen sisältö on saatu käyttäjän näkyville.

Mobiilisivustojen toteutuksessa kannattaa harkita käytettäväksi esimerkiksi Googlen AMP (*Accelerated Mobile Pages*)-tekniikkaa². AMP-tekniikka tarjoaa erityisesti uutissivustoille suunnitellun optimoidun HTML-osajoukon ja JavaScript-kirjaston, joiden tavoitteena on saada sisältö näkymään käyttäjille mahdollisimman nopeasti. AMP-tekniikan avulla pystyy helposti noudattamaan useita hyviä performanssikäytäntöjä ja esimerkiksi sivun yläosan latautumisenopeutta priorisoidaan automaattisesti. AMP-tekniikka asettaa kuitenkin paljon rajoitteita sivun sisällölle ja mainonnalle, joten kaikkiin tapauksiin se ei ole käyttökelpoinen.

Taustajärjestelmissä tulee kiinnittää huomiota vasteaikoihin. Rajapinnan vastaukset tulee tallentaa välimuisteihin ja välttää syvälle taustajärjestelmiin asti meneviä transaktioita. Rajapinnat tulee myös suunnitella siten, että päätelaitteiden tarvitsee tehdä mahdollisimman vähän erillisiä kyselyitä ja että vastauksissa siirretään vain päätelaitteen kannalta olennaista dataa.

²Accelerated Mobile Pages Project <https://www.ampproject.org/>

6.2 Natiivisovellusten toteutusvaihtoehdot

Kaikkien uutispalveluiden tapaukseen sopivaa yksiselitteistä suositusta natiivisovellusten toteuttamisteknologiaksi ei pystytty työn tulosten perusteella tunnistamaan. Sen sijaan työn tuloksista pystyttiin muodostamaan periaatteet, joiden perusteella valinnan hybridi-lähestymistavan hyödyntämisen ja täysin natiivitekniikoihin pohjautuvan toteutuksen välillä voi tehdä.

6.2.1 Hybridi mahdollistaa yhteneväisen sisällön ja ulkoasun, helpon hallittavuuden sekä tehokkaan kehittämisen usealle alustalle

Hybridi-lähestymistapaa kannattaa hyödyntää natiivisovellusten toteuttamisessa silloin, kun pyritään *mahdollisimman yhteneväiseen sisältöön ja ulkoasuun eri alustojen välillä*. Hybridi mahdollistaa uutispalvelun toimitukselle tarkan kontrollin palvelun sisältöön ja ulkoasuun kaikissa kanavissa.

Hybridi-lähestymistavalla pystytään helposti noudattamaan monia edellä esitettyjä strategian yleisiä periaatteita. Esimerkiksi sisällön rakenteen ja käyttöliittymälogiikan määrittely taustajärjestelmissä on hybriditoteutuksessa luonnollista. Hybridi-lähestymistavassa pystytään jakamaan ohjelma-logiikkaa ja ulkoasua paitsi eri alustojen, niin myös web-sivuston kanssa, mikä mahdollistaa tehokkaan kehitystyön usean alustan sovellusten ja sivuston välillä.

Hybridi-näkymien käyttö sovelluksessa kannattaa kuitenkin rajata varsinaisiin sisältönäkyymiin, kuten etusivuun, osiosivuihin ja artikkelisivuihin. Sovellusten ylätasen käyttöliittymä, kuten valikot ja siirtymät eri näkymien välillä, kannattaa edelleen toteuttaa natiivisti käyttäen alustakohtaisia komponentteja sekä noudattaen alustan käyttöliittymäkonventioita. Tämän avulla varmistetaan riittävä alustanmukainen käyttökokemus ja mahdollisuudet tehokkaaseen kokonaisuuden hallintaan.

6.2.2 Natiivi mahdollistaa alustakohtaisen mukauttamisen ja käyttökokemuksen

Täysin natiivitekniikoihin pohjautuvaa toteutusta kannattaa hyödyntää silloin, kun *palvelun sisältöä ja ulkoasua halutaan mukauttaa alustakohtaisesti*. Alustakohtainen mukauttaminen mahdollistaa alustan käyttökonventioiden

tarkan noudattamisen ja alustalle suunnitellun parhaan käyttökokemuksen. Usealle alustalle kehitettäessä työn määrä täysin natiivitekniikoihin pohjautuvassa lähestymistavassa on todennäköisesti hybridi-lähestymistapaa suurempi, mutta edellä esitettyjä yleisiä periaatteita seuraamalla myös natiivisovelluksista on mahdollista tehdä yhteneväinen ja hallittava kokonaisuus. Erityisesti kannattaa kiinnittää huomiota rajapintojen suunnitteluun siten, että mahdollisimman paljon ohjelmalogiikkaa voidaan käsitellä taustajärjestelmissä.

6.3 Yhteenveto

Mobiiliteknologiastrategian tärkein yleinen periaate on, että mahdollisimman suuri osa sisällön rakenteesta ja käyttöliittymälogiikasta kannattaa määritellä palvelun taustajärjestelmissä, eikä eri jakelukanavissa. Tällöin eri kanavat saadaan pidettyä yhteneväisempinä ja muutosten tekeminen on helpompaa ja nopeampaa. Muita strategian yleisiä periaatteita ovat:

- Sekä mobiililaitteille optimoitu web-sivusto, että alustojen natiivisovellukset tarvitaan
- Jaettua ohjelmalogiikkaa eri palveluiden ja alustojen välillä kannattaa suosia
- Elinkaariajattelu: natiivisovellusten päivitysilmoitukset, sulkemiseen varautuminen ja rajapintojen versiointi
- Palveluiden toimintanopeuden ja laadun korostaminen

Natiivisovellusten toteutustekniikan valitsemiseen täyden natiivitoteutuksen ja hybridi-lähestymistavan välillä löydettiin tutkimuksessa yksinkertainen pääperiaate: Jos palvelulle halutaan mahdollisimman yhteneväinen sisältö ja ulkoasu kaikissa alustoissa, niin hybridi-lähestymistavan hyödyntäminen on paras vaihtoehto. Jos puolestaan halutaan mukauttaa palvelun sisältöä ja ulkoasua alustakohtaisesti, niin täysin natiivitekniikoihin perustuva toteutus on parempi lähestymistapa.

Luku 7

Keskustelu ja arviointi

Tässä luvussa arvioidaan työn tuloksia ja sitä, miten ne liittyvät laajempaan kontekstiin. Luvussa keskustellaan myös mobiiliteknologioiden kehitymisestä ja annetaan ideoita tuleviin tutkimuksiin.

7.1 Työn tulosten arviointi

Tämän diplomityön tapaustutkimuksena on käsitelty Ilta-Sanomien mobiilipalveluita. Arvioitaessa työn tulosten yleistettävyyttä muihin uutispalveluihin on huomiotava Ilta-Sanomien erityispiirteet iltapäivälehtenä, mainosrahoitteisena palveluna sekä Suomen suurimpana uutispalveluna.

Osa työn tuloksista saattaa liittyä erityisesti iltapäivälehtien genreen. Esimerkiksi työssä identifioitu vaatimus sisällön ulkoasun tarkalle välittymiselle kaikkiin kanaviin saattaa olla vähemmän kriittinen muun tyyppisille uutispalveluille. Suurin osa havaituista vaatimuksista ja teknisten toteutusvaihtoehtojen ominaisuuksista ei kuitenkaan liity erityisesti iltapäivälehtien esitystapaan, jolloin ne ovat yleistettävissä kaikkiin journalistisiin uutispalveluihin.

Osa työssä identifioituista uutispalveluiden vaatimuksista mobiiliteknologioille ja esitetystä mobiiliteknologiastrategiasta voi olla yleistettävissä myös muihin kuin uutispalveluihin, esimerkiksi sosiaalisen median palveluihin.

Arvioitaessa työn tulosten ajankohtaisuutta ja vanhentumista tulevaisuudessa tulee erotella työn tulokset kahteen osaan: uutispalveluiden vaatimukset mobiiliteknologioille sekä mobiiliteknologiastrategian yleiset periaatteet pysyvät todennäköisesti relevantteina pidempään kuin natiivi-, web- ja hybridi-

teknologioiden ominaisuudet ja niiden vastaaminen uutispalvelun vaatimukseen.

Työssä identifioitujen uutispalvelun vaatimukset mobiiliteknologioille ovat suurimmalta osin teknologiariippumattomia ja todennäköisesti sovellettavissa myös tulevaisuuden toteutusteknologioilla. Myös esitetyt mobiiliteknologiastrategian yleiset periaatteet ovat melko korkealla abstraktiotasolla, jolloin ne saattavat säilyä relevantteina tulevaisuudessakin. Esimerkiksi logiikan keskittäminen taustajärjestelmiin ja elinkaariajattelu ovat periaatteita jotka todennäköisesti säilyvät relevantteina pitkään.

7.2 Mobiiliteknologioiden ja markkinatilanteen kehittyminen

Mobiiliteknologiat ja mobiilipalveluiden toteutusvaihtoehdot kehittyvät hyvin nopeasti. Jo tämän diplomityön tekemisen aikana sekä mobiilialustat että niiden ohjelmistokehitystyökalut ovat kehittyneet paljon eteenpäin. Esimerkiksi Apple on julkaissut *Swift*-ohjelmointikielen¹ ja myöhemmin tehnyt siitä avointa lähdekoodia², jotta se leviäisi käyttöön mahdollisimman laajasti.

Usealle alustalle tehtävää kehitystyötä helpottamaan on puolestaan julkaistu useita uusia teknologioita, kuten Facebookin *React Native*³. React Nativen tarkoitus ei ole ensisijaisesti jakaa koodia suoraan alustojen välillä, vaan mahdollistaa JavaScript-pohjainen kehitys usealle alustalle samalla teknologialla *“learn once, write anywhere.”* -periaatteella. (Occhino, 2015)

Microsoft puolestaan tarjoaa vaihtoehdon usealle mobiilialustalle kehittämiseen ostamansa *Xamarin*⁴-alustan avulla. Xamarin pohjautuu *C#*-kieleen ja koodin ristiinkääntämiseen (engl. *cross-compiling*) alustojen natiivisovelluksiksi iOS:lle, Androidille ja Windows Phonelle. Microsoft on ilmoittanut julkaisemansa Xamarinin avoimena lähdekoodina ja ilmaiseksi kehittäjien käyttöön kaikilla alustoilla. (Friedman, 2016)

Myös web-pohjaisten mobiilipalveluiden osalta tapahtuu merkittävää kehitystä, esimerkiksi Google kehittää *progressive web appseja*⁵, jotka on suunniteltu toimimaan myös offline-käytössä. Google kehittää myös yhteis-

¹<https://developer.apple.com/swift/>

²<https://swift.org/>

³<https://facebook.github.io/react-native/>

⁴<https://xamarin.com/>

⁵<https://developers.google.com/web/progressive-web-apps>

työssä useiden uutispalveluiden kanssa AMP (*Accelerated Mobile Pages*)-tekniikkaa⁶, jonka tarkoituksena on tehdä mobiiliwebistä nopeampi käyttää ja uutisten lukemisesta miellyttävämpää.

Mobiilialustojen markkinatilanne on myös kehittynyt työn tekemisen aikana. Markkinaa hallitsee⁷ entistä vahvemmin kaksi alustaa: Googlen Android ja Applen iOS. Microsoftin Windows Phonen markkina-asema on heikentynyt myös Suomessa ja uudet mobiilialustat kuten Firefox OS ja Jollan Sailfish eivät ole saavuttaneet menestystä. Markkinatilanteen kehittyminen vaikuttaa myös sovelluskehitykseen ja uudet sovellukset kehitetään nykyään usein vain suosituimmille alustoille Androidille ja iOS:lle.

Tuettavien alustojen määrän rajoittuminen vähentää alustakohtaisten natiivisovellusten kehittämisen kokonaistyötä. Tästä huolimatta ongelmakenttä eri alustojen pitämisestä yhteneväisenä ja kehitystyön tehokkuudesta säilyy, joten tekniset vaatimukset ja ratkaisut pysyvät relevantteina.

7.3 Tulevat tutkimukset

Työn aihepiiristä on olemassa melko vähän akateemista tutkimusta. Niinpä aiheita tuleville tutkimuksille riittää vielä paljon. Tämän työn yhdessä tunnistettiin muutamia ehdotuksia tulevien tutkimusten aiheiksi.

Erityisesti hybridi-sovellusten aihepiiri on vähän tutkittu. Esimerkiksi hybridi-lähestymistavan vaikutukset käytettävyyteen ja käyttökokemukseen on mielenkiintoinen aihe tutkittavaksi. Lisäksi kuten kappaleessa 5.3.4 todettiin, hybridi-sovellusten vaikutukset kaupallisiin tavoitteisiin jäivät tämän tutkimuksen puitteissa pääosin epäselviksi, joten niiden tutkiminen täydentäisi tämän työn tuloksia.

Hybridi-sovellusten lisäksi on olemassa muita teknisiä vaihtoehtoja toteuttaa natiivisovelluksia usealle alustalle ja jakaa toteutusta alustojen kesken. Tulevissa tutkimuksissa voisi tutkia kappaleessa 7.2 esiteltyjä uusia teknologioita, miten ne vertautuvat hybridi-sovelluksiin ja miten ne soveltuvat uutispalvelun vaatimuksiin. Voisivatko esimerkiksi React Native tai Xamarin sopia uutispalveluiden käyttötapaukseen? Nämä teknologiat ovat niin uusia, ettei niistä ole vielä lainkaan akateemista tutkimusta.

⁶<https://www.ampproject.org/>

⁷<http://www.gartner.com/newsroom/id/3169417>

Luku 8

Yhteenveto

Tämän tutkimuksen tarkoituksena oli selvittää minkälaisilla mobiiliteknologioilla pystytään parhaiten vastaamaan uutispalvelun vaatimuksiin. Tutkimuksessa vastattiin kolmeen tutkimuskysymykseen:

1. Minkälaisia vaatimuksia uutispalvelulla on käytettäville mobiiliteknologioille?

Tutkimuksessa kerätyistä materiaaleista identifioitiin viisi päävaatimusta mobiilille uutispalvelulle ja sen toteusteknologioille: *yhteneväisyys, muokattavuus, käyttökokemus ja ulkoasu, kaupalliset tavoitteet sekä nopeus ja luotettavuus*. Lisäksi identifioitiin yksi tulevaisuuden vaatimus, *personoitavuus*.

Yhteneväisyys tarkoittaa, että palvelun sisällön tulee toimia kaikissa kanavissa yhteneväisenä ja helposti hallittavana kokonaisuutena. Muokattavuus puolestaan edellyttää, että teknologiat mahdollistavat nopeat ja hallitut muutokset palvelun sisältöön kaikissa kanavissa. Käyttökokemuksen ja ulkoasun osalta vaatimus on, että palvelu tuntuu brändin mukaiselta kaikissa kanavissa, mutta noudattaa samalla mobiilialustojen käyttöliittymäkonventioita.

Kaupalliset tavoitteet tarkoittavat mainostuottojen takaamista mainosinventaaaria kasvattamalla: laajalla käyttäjäpohjalla, säännöllisellä käytöllä ja uusilla mainostyypeillä. Nopeuden ja luotettavuuden vaatimus edellyttää, että julkaistun sisällön tulee olla nopeasti käyttäjien saatavilla, palvelun käytön olla nopeaa ja palvelun toimia kaikissa tilanteissa luotettavasti.

Tulevaisuuden vaatimus palvelun personoitavuudesta tarkoittaa, että teknologioiden tulee todennäköisesti mahdollistaa palvelun sisällön personointi käyttäjäkohtaisesti tai kontekstuaalisesti.

2. Miten web- ja natiiviteknologiat vastaavat uutispalvelun vaatimukseen ja minkälaisia ongelmia niiden kanssa nykyisellään on?

Web- ja natiiviteknologioiden vastaaminen tutkimuksessa identifioituihin uutispalvelun vaatimukseen on tiivistetty taulukossa 8.1. Tutkimuksessa todettiin, että molemmilla teknologioilla on etunsa ja ongelmansa uutispalvelun toteuttamisessa.

Natiivisovellukset ja web-sivusto ovat uutispalvelun jakelukanavina erilaisia, mutta toisiaan täydentäviä ja tukevia. Web-sivusto on helposti saavutettavissa ilman asentamista ja mahdollistaa palveluun linkittämisen esimerkiksi sosiaalisesta mediasta. Natiivisovellukset puolestaan tarjoavat käyttäjille alustakohtaisen käyttökokemuksen ja monipuolisempia ominaisuuksia.

Mobiililaitteille optimoitu web-sivusto nähdään välttämättömyytenä, joka uutispalvelulla tulee joka tapauksessa olla. Natiivisovellukset puolestaan edistävät säännöllistä käyttöä ja ovat kaupallisesti tuottoisampia käyttäjää kohden, joten niihin investoiminen on myös järkevää.

3. Voiko hybridi-sovelluksilla yhdistää web-sivuston ja natiivisovellusten hyvät puolet ja vastata entistä paremmin mobiilin uutispalvelun vaatimukseen?

Tutkimuksessa todettiin, että hybridi-sovelluksien avulla on osittain mahdollista tuoda web-teknologioiden etuja natiivisovelluksiin ja siten vastata paremmin joihinkin uutispalvelun vaatimukseen. Toisaalta joihinkin asioihin hybridi-teknologiat tuovat lisähaasteita. Tutkimuksessa esitetään, että koko uutissovellusta ei kannata tehdä hybridinä, ainoastaan vaihtuvaa toimituksellista sisältöä näyttävät näkymät.

Hybridi-lähestymistavalla voidaan paremmin vastata uutispalvelun vaatimukseen sisällön yhteneväisyydestä sekä muokattavuudesta. Sisältö saadaan helpommin toteutettua yhteneväiseksi eri alustojen välillä ja muutoksia on helpompi tehdä. Käyttökokemuksen ja alustakohtaisen ulkoasun osalta hybridi on haasteellisempi. Vaikutukset kaupallisiin tavoitteisiin puolestaan ovat vielä epäselviä ja vaatisivat jatkoselvitystä. Nopeuden ja luotettavuuden osalta hybridi-sovellukset pääsevät melko lähelle natiivisovelluksia.

Yhteenveto hybridi-sovellusten vastaamisesta uutispalvelun vaatimukseen on myös esitetty taulukossa 8.1.

Uutispalvelun vaatimus	Natiivi-sovellukset	Web-sivusto	Hybridi-sovellukset
Sisällön yhteneväisyys, toimivuus ja hallittavuus	Hankalampia pitää yhteneväisenä, koska joudutaan toteuttamaan erikseen jokaiselle alustalle.	Yhtenevä sisältö kaikissa laitteissa. HTML-pohjainen sisältö toimii hyvin.	Eri alustojen sovellusten sisältö saadaan helpommin yhteneväiseksi ja hallittavaksi.
Muokattavuus	Muutokset vaativat uuden sovellusversion lataamista. Vanhat versiot aiheuttavat ongelmia, kun kaikki käyttäjät eivät päivitä sovelluksiaan.	Muutosten tekeminen nopeaa ja hallittua. Päivittyvät välittömästi kaikille käyttäjille. Mahdollistaa helpon AB-testauksen.	Mahdollistaa web-sivuston kaltaiset nopeat muutokset sisältöön, mutta suuremmat muutokset on tehtävä natiiveina päivityksinä.
Käyttökokemus ja ulkoasu	Paras mahdollinen alustan mukainen käyttökokemus.	Ei mahdollista saavuttaa alustan konventioiden mukaista käyttökokemusta.	Yhdistelmä natiivia ja web-käyttökokemusta, voi olla ongelmallinen.
Kaupalliset tavoitteet	Käyttäjää kohden tuottoisampia: käytetään säännöllisemmin, kalliimpia mainospaikkoja.	Mahdollistaa laajan käyttäjäpohjan palvelemisen tehokkaasti.	Web-mainokset toimivat paremmin. Muutoin vaikutukset epäselviä.
Nopeus ja luotettavuus	Push-notifikaatiot tuettuja. Sisällön esilataus ja offline-tuki helppo toteuttaa. Luotettavampi heikkolaatuisissa verkkoyhteyksissä.	Sisällön esilataus ja offline-tuki hankala toteuttaa. Huonompi tuki heikkolaatuisille verkkoyhteyksille.	Push-notifikaatiot tuettuja. Sisällön esilataus ja offline-tuki mahdollista toteuttaa. Suorituskyky lähellä natiivia.

Taulukko 8.1: Yhteenvedo tutkimuksessa vertailtujen teknologioiden vastaisesta identifiointiin uutispalvelun vaatimuksiin

Uutispalvelun mobiiliteknologiastrategia

Tärkein tutkimuksessa annettu yleinen strateginen periaate on, että mahdollisimman suuri osa sisällön rakenteesta ja käyttöliittymälogiikasta kannattaa määritellä palvelun taustajärjestelmissä, eikä eri jakelukanavissa. Tällöin eri kanavat saadaan pidettyä yhteneväisempinä ja muutosten tekeminen on helpompaa ja nopeampaa. Muita strategian yleisiä periaatteita ovat:

- Sekä mobiililaitteille optimoitu web-sivusto, että alustojen natiivisovellukset tarvitaan
- Jaettua ohjelmalogiikkaa eri palveluiden ja alustojen välillä kannattaa suosia
- Elinkaariajattelu: natiivisovellusten päivitysilmoitukset, sulkemiseen varautuminen ja rajapintojen versiointi
- Palveluiden toimintanopeuden ja laadun korostaminen

Natiivisovellusten toteutustekniikan valitsemiseen täyden natiivitoteutuksen ja hybridi-lähestymistavan välillä löydettiin tutkimuksessa yksinkertainen pääperiaate: Jos palvelulle halutaan mahdollisimman yhteneväinen sisältö ja ulkoasu kaikissa alustoissa, niin hybridi-lähestymistavan hyödyntäminen on paras vaihtoehto. Jos puolestaan halutaan mukauttaa palvelun sisältöä ja ulkoasua alustakohtaisesti, niin täysin natiivitekniikoihin perustuva toteutus on parempi lähestymistapa.

Lähteet

- Rami Aamulehto. Digitaalisten aikakaus- ja sanomalehtien julkaiseminen HTML5-tekniikalla. Master's thesis, Tietotekniikan koulutusohjelma, Aalto yliopiston Perustieteiden korkeakoulu, Espoo, Suomi, 2013. <http://urn.fi/URN:NBN:fi:aalto-201306146495>.
- Rahul C. Basole and Jürgen Karla. Value transformation in the mobile service ecosystem: A study of app store emergence and growth. *Serv. Sci.*, 4(1): 24–41, March 2012. ISSN 2164-3962. URL <http://dx.doi.org/10.1287/serv.1120.0004>.
- Tim Berners-Lee. Long Live the Web: A Call for Continued Open Standards and Neutrality. *Scientific American*, December 2010. URL <http://www.scientificamerican.com/article/long-live-the-web/>.
- Piers R. J. Campbell and Faheem Ahmed. An assessment of mobile os-centric ecosystems. *J. Theor. Appl. Electron. Commer. Res.*, 6(2):50–62, August 2011. ISSN 0718-1876. URL <http://dx.doi.org/10.4067/S0718-18762011000200006>.
- Andre Charland and Brian Leroux. Mobile application development: web vs. native. *Commun. ACM*, 54:49–53, May 2011. ISSN 0001-0782. URL <http://doi.acm.org/10.1145/1941487.1941504>.
- Jussi-Pekka Erkkilä. Web and Native Technologies in Mobile Application Development. Master's thesis, Department of Computer Science and Engineering, Aalto University School of Science, Espoo, Finland, 2013. <http://urn.fi/URN:NBN:fi:aalto-201304261933>.
- Brian Fling. *Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps*. O'Reilly Media, 2009. ISBN 978-0-596-15544-5.

- Nat Friedman. Xamarin for Everyone, 2016. URL <https://blog.xamarin.com/xamarin-for-all/>. Viitattu 17.4.2016.
- Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, 1967.
- Akhilesh Gupta. Mobile A/B testing at LinkedIn: how members shape our apps, 2014. URL <https://engineering.linkedin.com/mobile/mobile-ab-testing-linkedin-how-members-shape-our-apps>. Viitattu 11.1.2016.
- Henning Heitkötter, Sebastian Hanschke, and Tim A. Majchrzak. Evaluating cross-platform development approaches for mobile applications. In José Cordeiro and Karl-Heinz Krempels, editors, *Web Information Systems and Technologies*, volume 140 of *Lecture Notes in Business Information Processing*, pages 120–138. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36607-9. URL http://dx.doi.org/10.1007/978-3-642-36608-6_8.
- A. Juntunen, E. Jalonon, and S. Luukkainen. Html 5 in mobile devices – drivers and restraints. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 1053–1062, Jan 2013. doi: 10.1109/HICSS.2013.253.
- Kansallinen Mediatutkimus. Lähes puolet suomalaisista lukee sanomatai aikakauslehteään viikoittain mobiililaitteilla - miehet monilaitteikäytön edelläkävijöinä. Lehdistötiedote, 2014. URL http://www.sanomalehdet.fi/files/4008/KMT_lukijatiedote_helmikuu_2014.pdf. Viitattu 11.1.2016.
- Kansallinen Mediatutkimus. Aktiivi-ikäiset suomalaiset seuraavat lehtien sisältöä monipuolisesti kaikilla mahdollisilla välineillä. Lehdistötiedote, 2015. URL <http://mediaauditfinland.fi/kmt-2015-lukijatiedote/>. Viitattu 25.4.2016.
- Kimmo Karhu, Tingan Tang, and Matti Hämäläinen. Analyzing competitive and collaborative differences among mobile ecosystems using abstracted strategy networks. *Telemat. Inf.*, 31(2):319–333, May 2014. ISSN 0736-5853. URL <http://dx.doi.org/10.1016/j.tele.2013.09.003>.
- Adam Lella, Andrew Lipsman, and Ben Martin. The 2015 U.S. Mobile App Report. Comscore., 2015. URL <https://www.comscore.com/Insights/Presentations-and-Whitepapers/2015/The-2015-US-Mobile-App-Report>. Viitattu 11.1.2016.

- Tommi Mikkonen and Antero Taivalsaari. Apps vs. open web: The battle of the decade. In *MSE'2011 (Santa Monica, California, USA)*, pages 22–26, October 2011.
- Tommi Mikkonen and Antero Taivalsaari. Cloud computing and its impact on mobile software development: Two roads diverged. *Journal of Systems and Software*, 86(9):2318 – 2320, 2013. ISSN 0164-1212. URL <http://dx.doi.org/10.1016/j.jss.2013.01.063>.
- Miro Nieminen. Fallback Mechanisms for Connection Loss in Single-Page Web Applications. Master’s thesis, Department of Pervasive Computing, Faculty of Computing and Electrical Engineering, Tampere University of Technology, Tampere, Finland, 2015. <http://urn.fi/URN:NBN:fi:tty-201504231233>.
- Tom Occhino. React Native: Bringing modern web techniques to mobile, 2015. URL <https://code.facebook.com/posts/1014532261909640/react-native-bringing-modern-web-techniques-to-mobile/>. Viitattu 17.4.2016.
- Sanna Ottka. Comparison of mobile application development tools for multi-platform industrial applications. Master’s thesis, Department of Computer Science and Engineering, Aalto University School of Science, Espoo, Finland, 2015. <http://urn.fi/URN:NBN:fi:aalto-201505132661>.
- Guy Podjarny. iOS Browsers Speed Bakeoff, 2013. URL <http://www.guypo.com/ios-browsers-speed-bakeoff/>. Viitattu 5.1.2016.
- Kimmo Puputti. Mobile HTML5: Implementing a Responsive Cross-Platform Application. Master’s thesis, Department of Computer Science and Engineering, Aalto University School of Science, Espoo, Finland, 2012. <http://urn.fi/URN:NBN:fi:aalto-201210043228>.
- Tuure Savuoja. Hybridisovellukset mobiilisovelluskehityksessä. Master’s thesis, Tietotekniikan koulutusohjelma, Aalto-yliopiston Perustieteiden korkeakoulu, Espoo, Suomi, 2015. <http://urn.fi/URN:NBN:fi:aalto-201506303450>.
- Anselm L. Strauss and Juliet M. Corbin. *Basics of qualitative research: grounded theory procedures and techniques*. Sage Publications, 1990. ISBN 9780803932500.
- Antero Taivalsaari and Tommi Mikkonen. From apps to liquid multi-device software. *Procedia Computer Science*, 56:34 – 40, 2015. ISSN 1877-0509.

- URL <http://dx.doi.org/10.1016/j.procs.2015.07.179>. The 10th International Conference on Future Networks and Communications (FNC 2015) / The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015) Affiliated Workshops.
- TJ VanToll. Why iOS 8's WKWebView is a Big Deal for Hybrid Development, 2014. URL <http://developer.telerik.com/featured/why-ios-8s-wkwebview-is-a-big-deal-for-hybrid-development/>. Viitattu 5.1.2016.
- Seppo Varjus. Ilta-Sanomat on Suomen suurin uutismedia, 2015. URL <http://www.iltasanomat.fi/kotimaa/art-2000000993708.html>. Viitattu 25.4.2016.
- W.I. Warburton. *Towards a grounded theory of computer-assisted assessment uptake in UK universities*. PhD thesis, University of Southampton, March 2006.
- Juha Winter. *Success Factors of Mobile Business Ecosystems : From Hardware-Centric to Content and Advertising Based Business Models*. Licentiate thesis, Department of Computer Science and Engineering, Aalto University School of Science and Technology, Espoo, Finland, 2014. <http://urn.fi/URN:NBN:fi:aalto-201411193036>.
- Robert K. Yin. *Case Study Research: Design and Methods*. Applied Social Research Methods. SAGE Publications, 2003. ISBN 9780761925521.
- Anton Zvonkov. Choosing the implementation technology for content-driven mobile service. Master's thesis, Department of Computer Science and Engineering, Aalto University School of Science, Espoo, Finland, 2013.

Liite A

Haastattelukysymykset

A.1 Kysymykset kaikille

Taustatiedot

- Mikä on oma roolisi Ilta-Sanomien digitaalisten palveluiden kehittämisessä, erityisesti mobiilipalveluiden osalta?

Mobiili uutispalvelu

- Minkälainen on hyvä mobiili uutispalvelu?
- Miten mainitsemasi asiat toteutuvat Ilta-Sanomissa?
- Minkälaisia erityispiirteitä uutispalvelulla on verrattuna muihin palveluihin / sovelluksiin?

Mobiilipalveluiden teknologiavalinnat

- Minkälaisia mobiilipalveluiden teknologiavalintoja Ilta-Sanomien palveluihin ollaan tehty ja mihin tekijöihin ne ovat pohjautuneet?
- Mitkä ovat tärkeimmät opitut asiat valituista teknologioista? Mikä on toiminut hyvin tai huonosti?

Hybridisovellukset

- Mitkä ovat mielestäsi suurimmat edut joita hybridisovelluksilla voitaisiin saavuttaa?

- Mitkä ovat hybridisovellusten riskit ja huonot puolet?

Tulevaisuuden ajatukset

- Minkälaisia (uutis-) mobiilipalveluita käytämme viiden vuoden päästä? Entä kymmenen vuoden päästä?
- Mitä isoja muutoksia seuraavan vuoden tai kahden aikana tullaan konkreettisesti tekemään?

A.2 Lisäkysymykset Sanoman edustajille

Taustatiedot

- Minkälaisia mobiilipalveluita Ilta-Sanomilla on vuosien varrella tehty?

Liiketoimintanäkökulma

- Miten valitaan minkälaisiin mobiilipalveluihin halutaan investoida ja lähteä toteuttamaan? Minkälaiset tekijät vaikuttavat investointipäätökseen?
- Mitkä mobiilipalvelut ovat olleet parhaita investointeja?

Kehityssykli, kehitysnopeus

- Minkälaisissa sykleissä palveluita kehitetään? Eroaako nykytilanne ideaalitalanteesta?
- Minkälaisia muutoksia palveluihin tulee niiden elinkaaren aikana?
- Kuinka nopeasti muutokset palveluihin olisi hyvä saada tehtyä?

Uutispalvelun mukauttaminen mobiiliin

- Pitäisikö uutispalvelun eri kanavissa esittää sama sisältö vai mukauttaa sisältö kanavakohtaisesti?
- Kumpi on parempi vaihtoehto Ilta-Sanomien etusivun osalta: A) Kaikkiin kanaviin toteutetaan sisällöltään identtinen etusivu, jota toimitus käsin hallitsee. B) Etusivun sisältö mukautetaan eri kanaviin, joko automaattisesti tai käsin halliten.

- Ulkoasu: pitäisikö etusivun, osioiden ja artikkeleiden ulkoasu olla mahdollisimman samanlainen kaikissa kanavissa, vai mukautetaanko näyttöasua kanavakohtaisesti?

Web-sivustot ja natiivisovellukset jakelukanavina

- Mihin eri käyttötapauksiin web-sivustot ja natiivisovellukset vastaavat?
- Jos saisit valita vain sivuston tai alustakohtaiset sovellukset, kumman valitsisit?
- Onko sivustoilla ja sovelluksilla eroa esimerkiksi käyttöönottokynnyksessä, käytön säännöllisyydessä tai muussa vastaavassa?

A.3 Lisäkysymykset kehittäjille

Mobiilipalveluiden teknologiavalinnat

- Mitä hyviä teknologiaratkaisuja on tehty kehittämissäsi palveluissa?
- Mitä on mennyt huonosti? Mitä tekisit toisin jos nyt voisit muuttaa aikaisempia valintoja?
- Mitkä ovat tärkeimmät oppimasi asiat?

Kehityssyklit, kehitysnopeus

- Kuinka nopeasti kehittämääsi palveluun on mahdollista tehdä muutoksia?

Dynaaminen etusivu

- Miten voidaan toteuttaa sisällöltään ja rakenteeltaan dynaaminen etusivu?
- Olisiko natiivisti mahdollista saada aikaan samanlainen joustavuus kuin web-sivustoilla ja hybridissä?
- Mikä on oma ehdotuksesi dynaamisen etusivun toteuttamiseksi?